

**PENGEMBANGAN APLIKASI PENERIMAAN SISWA BARU
PADA YAYASAN PERGURUAN ISLAM AS'SAADAH
PONDOK KELAPA JAKARTA TIMUR**



Mahfud Ramdhani A

5235107392

**Skripsi ini Ditulis Untuk Memenuhi Sebagai Persyaratan dalam
Memperoleh Gelar Sarjana**

**Program Studi Pendidikan Teknik Informatika & Komputer
Jurusan Teknik Elektro**

Fakultas Teknik

Universitas Negeri Jakarta

2016

**PENGEMBANGAN APLIKASI PENERIMAAN SISWA BARU PADA
YAYASAN PERGURUAN ISLAM AS'SAADAH PONDOK KELAPA
JAKARTA TIMUR**

MAHFUD RAMDHANI A

ABSTRAK

Penerimaan Siswa Baru (PSB) merupakan kegiatan yang amat penting karena proses tersebut mempunyai nilai strategis guna menjaring calon siswa yang berkualitas. Sistem yang sedang berjalan dalam pendaftaran siswa baru di Yayasan Perguruan Islam (YPI) As Sa'adah masih manual dan pencarian data siswa sewaktu di perlukan membutuhkan waktu cukup lama.. Hal ini sangat diperlukan komputer sebagai alat bantu dalam suatu sistem informasi tersebut. Proses pendaftaran, penyeleksian dan pengelompokan kelas merupakan langkah-langkah yang ditempuh untuk proses penerimaan siswa baru di YPI Assa'adah. Kegiatan pendaftaran siswa baru bertujuan untuk proses pendaftaran yang terstruktur dan informatif, sehingga dapat dihasilkan proses kerja yang lebih baik dan maksimal dengan menggunakan penyimpanan data yang tersentralisasi yang dapat memudahkan kegiatan perubahan data, pencarian dan penyimpanan data dengan menggunakan teknologi komputer. Dalam penelitian ini menggunakan metode eksperimen. Aplikasi penerimaan siswa baru dikembangkan dengan Bahasa pemrograman *PHP* dan basis data *MySQL* dengan *XAMPP* sebagai *local server*. Metode yang digunakan dalam pengembangan sistem ini menggunakan metode pengembangan perangkat lunak *waterfall*. Pengujian aplikasi penerimaan siswa baru ini menggunakan metode *black box testing* untuk mengetahui sistem penerimaan siswa baru berjalan dengan baik. Hasil penelitian menyimpulkan bahwa aplikasi ini dapat di manfaatkan dalam proses penerimaan siswa baru di YPI As Sa'adah.

Kata kunci : aplikasi penerimaan siswa baru, *waterfall*, *black box testing*

**APPLICATION DEVELOPMENT OF NEW STUDENT ADMISSION IN
ISLAMIC EDUCATION FOUNDATION AS'SAADAH PONDOK KELAPA
JAKARTA TIMUR**

MAHFUD RAMDHANI A

ABSTRACT

New student admission is an important occasion due to the process has strategic value to select qualified student candidates. The ongoing system on new student admission is still manual. The data searching still costs big amount of time. The computer is needed to assist in certain information system. Admission process, selection, and class grouping are steps to gather admission data systematically. New student admission aims to be more structural and informative in order to achieve better and maximum work by using centralized data storage which can ease data changing and searching and saving data with computer. This study conducted experimental method. New student application was being developed using programming language PHP and database MySQL with XAMPP as local server. This system used waterfall software developing method. The new student admission application test used black box method to know running system well. The result of this study concluded that this system could be beneficial in new student admission process in Islamic As Sa'adah Foundation.

Key words : Application new student admission, waterfall, black box testing.

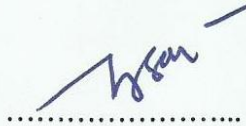
LEMBAR PENGESAHAN

NAMA DOSEN

TANDA TANGAN

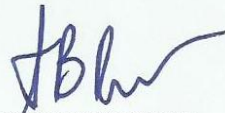
TANGGAL

Drs. Bachren Zaini, M.Pd
(Dosen Pembimbing I)



12/1 2016

Bambang Prasetya Adhi, S.Pd., M.Kom
(Dosen Pembimbing II)



12/1 2016


PENGESAHAN PANITIA UJIAN SKRIPSI

NAMA DOSEN

TANDA TANGAN

TANGGAL

Dr. Yuliatr Sastrawijaya, M.Pd
(Ketua Penguji)



12/1 2016

Hamidillah Ajie, MT
(Dosen Penguji)



12/1 2016

Prof. Dr. Ir. Ivan Hanafi, M.Pd
(Dosen Ahli)



11/1 2016

Tanggal Lulus: 6 Januari 2016

HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Karya tulis skripsi saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik sarjana, baik di Universitas Negeri Jakarta maupun di perguruan tinggi lain.
2. Karya tulis ini adalah murni gagasan, rumusan, dan penelitian saya sendiri dengan arahan dosen pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya yang berlaku di Universitas Negeri Jakarta.

Jakarta, Januari 2016

Yang membuat pernyataan

Mahfud Ramdhani A

5235107392

KATA PENGANTAR

Segala puji dan syukur kehadiran Allah swt atas rahmat dan karuniaNya yang telah diberikan kepada peneliti, sehingga peneliti dapat menyelesaikan skripsi dengan judul: “PENGEMBANGAN APLIKASI PENERIMAAN SISWA BARU DI YAYASAN PERGURUAN ISLAM AS’SAADAH PONDOK KELAPA”. Dalam merencanakan, menyusun, dan menyelesaikan penelitian skripsi, peneliti banyak menerima bantuan, bimbingan, dan motivasi serta dukungan dari berbagai pihak, oleh karena itu pada kesempatan ini peneliti bermaksud

mengucapkan terima kasih yang sebesar-besarnya disampaikan kepada:

1. Bapak Drs. Bachren Zaini, M.Pd selaku pembimbing I dan Bapak Bambang Prasetya Adhi, S.Pd., M.Kom selaku pembimbing II yang telah memberikan motivasi, arahan dan kepercayaan kepada peneliti dalam menyelesaikan skripsi ini.
2. Ibu Dr. Yuliatry Sastrawijaya, M.Pd selaku Kepala Program Studi Pendidikan Teknik Informatika dan Komputer
3. Orang tua yang tak hentinya memanjatkan doa dan memberikan semangat kepada peneliti dalam menyelesaikan skripsi ini.

Peneliti menyadari bahwa skripsi ini jauh dari kesempurnaan, untuk itu peneliti mohon maaf apabila terdapat kekurangan dan kesalahan baik dari isi maupun penelitian. Akhir kata, saya berharap semoga skripsi ini bermanfaat bagi ymag membacanya.

Jakarta, Januari 2016

Mafud Ramdhani A

DAFTAR ISI

| | |
|-----------------------------------------------------|-------------|
| ABSTRAK | i |
| ABSTRACT | .ii |
| LEMBAR PENGESAHAN. | iii |
| HALAMAN PERNYATAAN..... | iv |
| KATA PENGANTAR..... | v |
| DAFTAR ISI..... | vi |
| DAFTAR TABEL. | ix |
| DAFTAR GAMBAR..... | xi |
| DAFTAR LAMPIRAN..... | xiii |
| BAB I PENDAHULUAN | |
| 1.1. Latar Belakang Masalah. | 1 |
| 1.2. Identifikasi Masalah. | 3 |
| 1.3. Pembatasan Masalah. | 4 |
| 1.4. Perumusan Masalah. | 4 |
| 1.5. Tujuan Penelitian. | 4 |
| 1.6. Kegunaan Penelitian. | 5 |
| BAB II KAJIAN TEORITIS DAN KERANGKA BERPIKIR | |
| 2.1. Kerangka Teoritis..... | 6 |
| 2.1.1. Perangkat Lunak | 6 |
| 2.1.2. Pengertian Aplikasi. | 6 |
| 2.1.3. Framework..... | 7 |
| 2.1.4. MVC | 8 |
| 2.1.5. ORM | 12 |

| | |
|-------------------------------------------------------|----|
| 2.1.6. Pengertian Basis data..... | 13 |
| 2.1.7. DBMS..... | 13 |
| 2.1.8. <i>Unfied Modeling Language</i> | 14 |
| 2.1.9. Model Proses Pengembangan Perangkat Lunak..... | 23 |
| 2.1.9.1. Model Sekuensial Linear..... | 23 |
| 2.1.9.2. Model Prototipe..... | 26 |
| 2.1.9.3. Model RAD..... | 29 |
| 2.1.9.4. Model Spiral..... | 31 |
| 2.1.10. Pengujian..... | 34 |
| 2.1.10.1 <i>Black Box Testing</i> | 35 |
| 2.1.10.2 <i>White Box Testing</i> | 40 |
| 2.1.11. Penerimaan Siswa Baru..... | 42 |
| 2.2. Kerangka Berpikir..... | 43 |

BAB III METODOLOGI PENELITIAN

| | |
|----------------------------------------|----|
| 3.1. Tujuan Operasional..... | 45 |
| 3.2. Tempat dan Waktu Penelitian..... | 45 |
| 3.3. Metode Penelitian..... | 45 |
| 3.4. Alat dan Bahan Penelitian..... | 46 |
| 3.5. Langkah – Langkah Penelitian..... | 47 |
| 3.5.1 Analis..... | 47 |
| 3.5.2 Desain..... | 49 |
| 3.5.3 implementasi..... | 50 |
| 3.5.4 Pengujian..... | 50 |
| 3.5.5 Pemeliharaan..... | 57 |

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

| | |
|-----------------------------------------|-----|
| 4.1 Hasil Analisis dan Pembahasan. | 58 |
| 4.2 Perancangan | 60 |
| 4.2.1 <i>Usecase</i> | 60 |
| 4.2.2 <i>Sequence Diagram</i> | 63 |
| 4.2.3 <i>Class Diagram</i> | 76 |
| 4.2.4 Struktur <i>Database</i> | 88 |
| 4.2.5 Implementasi Hasil Tampilan. | 94 |
| 4.3 Pengujian..... | 104 |
| 4.4 Pemeliharaan..... | 111 |

BAB V KESIMPULAN DAN SARAN

| | |
|----------------------|-----|
| 5.1 Kesimpulan. | 112 |
| 5.2 Saran. | 112 |
| DAFTAR PUSTAKA. | 114 |
| LAMPIRAN..... | 115 |
| Tentang Penulis..... | 119 |

DAFTAR TABEL

| | |
|---------------------------------------------------------------------|----|
| Tabel 3.1 Kriteria Pengujian Kebutuhan Fungsional <i>User</i> | 52 |
| Tabel 3.2 Kriteria Pengujian Kebutuhan Fungsional Admin..... | 54 |
| Tabel 4.1 User Class. | 77 |
| Tabel 4.2 Casis clas..... | 78 |
| Tabel 4.3 Pembayaran PSB Class..... | 80 |
| Tabel 4.4 Calon Peserta Search Model Class. | 80 |
| Tabel 4.5 Angkatan Class. | 81 |
| Tabel 4.6 Unit Class..... | 81 |
| Tabel 4.7 Admin Class..... | 82 |
| Tabel 4.8 Admin Search Model Class. | 82 |
| Tabel 4.9 Username Class..... | 83 |
| Tabel 4.10 Pegawai Class. | 83 |
| Tabel 4.11 Siswa Class. | 84 |
| Tabel 4.12 Pengumuman Search Model Class. | 85 |
| Tabel 4.13 Pengumuman Class..... | 85 |
| Tabel 4.14 Setting Fitur Class..... | 86 |
| Tabel 4.15 Set Pembayaran Class | 86 |
| Tabel 4.16 login Form Class | 87 |
| Tabel 4.17 Setting Value Class | 87 |
| Tabel 4.18 Signup Form Class | 88 |
| Tabel 4.19 Tabel Admin. | 88 |
| Tabel 4.20 Tabel Calon Peserta | 89 |
| Tabel 4.21 Tabel <i>Migration</i> | 91 |

| | |
|---------------------------------------------------------|-----|
| Tabel 4.22 Tabel Pembayaran..... | .91 |
| Tabel 4.23 Tabel Pengumuman | .91 |
| Tabel 4.24 Tabel <i>Setting Fitur</i> | .92 |
| Tabel 4.25 Tabel <i>setting value</i> | .92 |
| Tabel 4.26 Tabel <i>User</i> | .93 |
| Tabel 4.27 Hasil Pengujian Fungsional <i>User</i> | 105 |
| Tabel 4.28 Tabel Hasil Pengujian Fungsional Admin. | 107 |

DAFTAR GAMBAR

| | |
|------------------------------------------------------------|----|
| 2.1. Cara Kerja MVC | 9 |
| 2.2. Model | 10 |
| 2.3. View | 11 |
| 2.4. Controller..... | 12 |
| 2.5. Simbol <i>Usecase</i> | 17 |
| 2.6. Simbol <i>Sequence Diagram</i> | 22 |
| 2.7. Model <i>Sequensial Linier</i> | 25 |
| 2.8. Model Prototipe. | 27 |
| 2.9. Model RAD. | 29 |
| 2.10. Model Spiral. | 33 |
| 2.11 Kerangka Berpikir..... | 44 |
| 4.1. <i>Usecase</i> Admin..... | 61 |
| 4.2. <i>Usecase</i> Calon Siswa. | 62 |
| 4.3. <i>Sequence Diagram</i> Create Akun Calon Siswa | 63 |
| 4.4. <i>Sequence Diagram</i> Login Siswa..... | 64 |
| 4.5. <i>Sequence Diagram</i> Daftar Diri Siswa..... | 65 |
| 4.6. <i>Sequence Diagram</i> Cetak Resi | 66 |
| 4.7. <i>Sequence Diagram</i> Lihat Kelas | 67 |
| 4.8. <i>Sequence Diagram</i> Login Admin | 68 |
| 4.9. <i>Sequence Diagram</i> Daftar Siswa Admin | 69 |
| 4.10. <i>Sequence Diagram</i> Cetak Laporan Admin..... | 70 |
| 4.11. <i>Sequence Diagram</i> Bagi Kelas Admin | 71 |
| 4.12. <i>Sequence Diagram</i> Setting Value Umur | 72 |

| | |
|---------------------------------------------------------|-----|
| 4.13. <i>Sequence Diagram</i> Setting Value Nilai | 73 |
| 4.14. <i>Sequence Diagram</i> Atur Fitur Admin | 74 |
| 4.15 <i>Sequence Diagram</i> Lihat Grafik Admin | 75 |
| 4.16 <i>Class Diagram</i> | 76 |
| 4.17 Beranda..... | 94 |
| 4.18 <i>Create Akun</i> | 94 |
| 4.19 <i>Form</i> Pendaftar | 95 |
| 4.20 Cetak Resi Pembayaran..... | 95 |
| 4.21 Cetak Biodata | 96 |
| 4.22 Pengumuman Seleksi | 97 |
| 4.23 <i>Form Login</i> | 97 |
| 4.24 <i>Form Login</i> Admin | 98 |
| 4.25 Halaman Pendaftaran Admin | 99 |
| 4.26 Calon Peserta pada Admin | 100 |
| 4.27 <i>Form</i> Registrasi | 100 |
| 4.28 Halaman Calon Siswa | 101 |
| 4.29 Halaman Pengumuman Informasi | 102 |
| 4.30 Kelola fitur..... | 102 |
| 4.31 Setting Value Kuota | 103 |
| 4.32 Setting Value Nilai..... | 104 |

DAFTAR LAMPIRAN

| | |
|------------------------------------------|-----|
| Lampiran 01 Hasil Tampilan Aplikasi..... | 115 |
| Lampiran 02 Surat Penelitian..... | 116 |
| Lampiran 03 Wawancara. | 117 |
| Lampiran 04 Pengujian Aplikasi. | 118 |

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan informasi teknologi saat ini semakin pesat, kebutuhan informasi dan pengolahan data dalam banyak aspek kehidupan manusia sangat penting. Dari perkembangan teknologi yang demikian pesat berdampak bagi seluruh kehidupan khususnya penyediaan informasi bagi suatu instansi atau perusahaan yang membutuhkan sistem pengelolaan data secara cepat, tepat dan akurat. Maka perlu menyusun suatu aplikasi pengelolaan data terkomputerisasi yang baik dan berdaya guna. Aplikasi tersebut diharapkan dapat menyelesaikan tugas-tugas rutin yang berguna dalam pengambilan keputusan oleh organisasi/instansi atau perusahaan tersebut.

Sistem informasi banyak digunakan dalam suatu lembaga. Misalnya dalam perusahaan, perkantoran dan pendidikan. Dalam lembaga pendidikan sistem informasi ini bertujuan untuk promosi atau memberikan gambaran umum mengenai profil lembaga pendidikan yang bersangkutan, administrasi, dan fasilitas yang lain. Sekolah merupakan lembaga pendidikan, yang menampung peserta didik dan dibina agar mereka memiliki kemampuan, kecerdasan dan keterampilan. Dalam proses pendidikan diperlukan pembinaan secara berkoordinasi dan terarah. Dengan demikian siswa diharapkan dapat mencapai prestasi belajar yang maksimal sehingga tercapainya tujuan pendidikan. Maka lembaga pendidikan di Yayasan

Perguruan Islam (YPI) Assa'adah Pondok Kelapa sampai saat ini masih menggunakan sistem informasi yang belum terkomputerisasi atau manual. Setiap calon siswa harus mengisi formulir pendaftaran diselembar kertas, dan panitia mencatat setiap data calon siswa kedalam buku besar dan menyimpan data dalam sebuah ruangan yang berisi dokumen – dokumen lain sehingga sering menimbulkan lambatnya suatu informasi yang dihasilkan atau harus ditingkatkan lagi terutama dalam bidang pendaftaran, penyeleksian dan pengelompokan kelas siswa baru di YPI As Sa'adah Pondok Kelapa. Penerimaan Siswa Baru (PSB) merupakan kegiatan yang amat penting karena proses tersebut mempunyai nilai strategis guna menjaring calon siswa yang berkualitas. Hal ini sangat diperlukan komputer sebagai alat bantu dalam suatu sistem informasi tersebut. Proses pendaftaran, penyeleksian dan pengelompokan kelas merupakan langkah-langkah yang ditempuh guna untuk mengumpulkan data pendaftaran secara sistematis.

Pendaftaran, penyeleksian dan pengelompokan kelas siswa baru di YPI As Sa'adah Pondok Kelapa masih tergolong menggunakan sistem penulisan, sehingga segala prosesnya dari mulai menginput data masih secara tertulis. Adapun sistem yang sedang berjalan dalam pendaftaran siswa baru di YPI As Sa'adah Pondok Kelapa masih menggunakan cara manual yaitu dengan mengisi formulir pendaftaran dan itu pun akan memperlambat pencarian siswa sewaktu data tersebut di perlukan. Sistem penerimaan siswa baru di YPI As Sa'adah Pondok Kelapa Pondok Kelapayang ditujukan untuk penyediaan informasi tentang data siswa baru yang akan di terima. Dengan komputer sebagai alat bantunya dalam menyelesaikan masalah yang dihadapi

oleh sekolah dan sudah waktunya memakai fasilitas komputer yang meringankan sedikit beban pekerjaan dan diharapkan dapat membantu YPI As Sa'adah Pondok Kelapa Pondok Kelapa untuk mencapai tujuan dalam waktu yang singkat. Dengan adanya aplikasi ini memudahkan *admin* atau panitia penerimaan siswa baru untuk proses seleksi dan mengolah data. Data pendaftaran nantinya akan berlanjut digunakan untuk pembayaran siswa, keperluan laporan siswa ke dinas pendidikan, dan laporan untuk Bantuan Operasional Sekolah (BOS) atau bantuan – bantuan lain.

Berdasarkan keadaan dan permasalahan tersebut, maka diperlukan Pengembangan Aplikasi Penerimaan Siswa Baru pada YPI As Sa'adah Pondok Kelapa.

1.2. Identifikasi Masalah

Berdasarkan latar belakang yang telah dikemukakan, maka yang menjadi perhatian dalam skripsi ini adalah:

1. Sistem pendaftaran siswa baru masih belum maksimal dengan dilakukannya mencatat setiap data calon siswa kedalam buku pendaftaran dan menyimpan data dalam sebuah ruangan yang berisi dokumen-dokumen lain sehingga tercampur. Akibatnya data terkadang salah atau bahkan tidak ada.

2. Sulitnya dalam pencarian data siswa baru, sehingga data atau informasi tentang siswa baru yang dihasilkan mengalami kekeliruan.
3. Laporan data siswa baru yang masih belum maksimal sehingga berpengaruh terhadap pengelompokan kelas dan laporan bantuan operasional sekolah.

1.3. Pembatasan Masalah

Berikut ini adalah batasan masalah yang diberikan dalam skripsi ini :

1. penerimaan siswa baru disusun hanya meliputi tentang penerimaan calon siswa baru
2. pada aplikasi ini tidak membahas keamanan sistem
3. sistem pembayaran hanya melalui kasir disekolah, belum menggunakan transfer via bank.

1.4. Perumusan Masalah

Berdasarkan latar belakang diatas, maka perumusan masalah dalam penelitian ini adalah “Bagaimana mengembangkan aplikasi penerimaan siswa baru pada YPI As Sa’adah Pondok Kelapa Pondok Kelapa agar terkomputerisasi dengan baik”?.

1.5. Tujuan Penelitian

Tujuan pembuatan aplikasi pendaftaran siswa baru pada YPI As Sa'adah Pondok Kelapa Pondok Kelapa ialah untuk melakukan kegiatan pendaftaran yang terstruktur dan informatif, sehingga dapat dihasilkan proses kerja yang lebih baik dan maksimal dengan menggunakan penyimpanan data yang tersentralisasi yang dapat memudahkan kegiatan perubahan data, pencarian dan penyimpanan data dengan menggunakan teknologi komputer

1.6. Manfaat Penelitian

Manfaat penelitian ini adalah :

1. Memudahkan calon siswa untuk melakukan proses pendaftaran tanpa harus melakukan interaksi langsung dengan bagian administrasi karena informasi tersebut dapat diperoleh dengan melakukan pencarian data melalui komputer yang terkoneksi secara internet.
2. Memudahkan bagian administrasi dikarenakan berkurangnya interaksi secara langsung dengan calon siswa.
3. Penyimpanan data yang terstruktur dikarenakan Sistem Pendaftaran siswa baru menggunakan *database* yang tersimpan didalam komputer.

BAB II

KAJIAN TEORITIS DAN KERANGKA BERPIKIR

2.1. Kajian Teoritis

2.1.1. Perangkat Lunak

Perangkat lunak adalah program computer yang berfungsi sebagai sarana interaksi antara pengguna dan perangkat keras. Perangkat lunak juga dapat dikatakan sebagai penerjemah perintah – perintah yang dijalankan pengguna computer untuk diteruskan atau diproses oleh perangkat keras. Perangkat lunak dibagi menjadi tiga tingkatan, yaitu tingkatan program aplikasi (*application program*) misalnya Microsoft Office, tingkatan sistem operasi (*operating system*) misalnya Microsoft Windows, tingkatan Bahasa pemrograman yang dibagi atas Bahasa pemrograman tingkat tinggi seperti pascal dan Bahasa pemrograman tingkat rendah, yaitu Bahasa rakitan (Janner Simartama, 2010). Perangkat lunak pada umumnya digunakan untuk mengontrol perangkat keras yang sering disebut *device driver*, melakukan proses perhitungan, berinteraksi dengan perangkat lunak yang lebih mendasar lainnya seperti sistem operasi, Bahasa pemrograman dan lain-lain.

2.1.2 Pengertian Aplikasi

Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti game, pelayanan masyarakat, periklanan, atau semua proses yang dilakukan manusia (Hengky W. Pranama, 2012). Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan

pekerjaan pekerjaan tertentu (Hendrayudi, 2008). Aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu (Yuhefizar, 2012). Aplikasi dapat digolongkan menjadi beberapa kelas yaitu perangkat lunak perusahaan (*enterprise*), perangkat lunak infrastruktur perusahaan, perangkat lunak informasi kerja, perangkat lunak media dan hiburan, perangkat lunak pendidikan, perangkat lunak pengembangan media, dan perangkat lunak rekayasa produk. Berdasarkan pengertian diatas, dapat disimpulkan bahwa aplikasi merupakan sebuah program yang dibuat dalam sebuah perangkat lunak untuk memudahkan pekerjaan atau tugas-tugas seperti penerapan, penggunaan, dan penambahan data yang dibutuhkan.

2.1.3 *Framework*

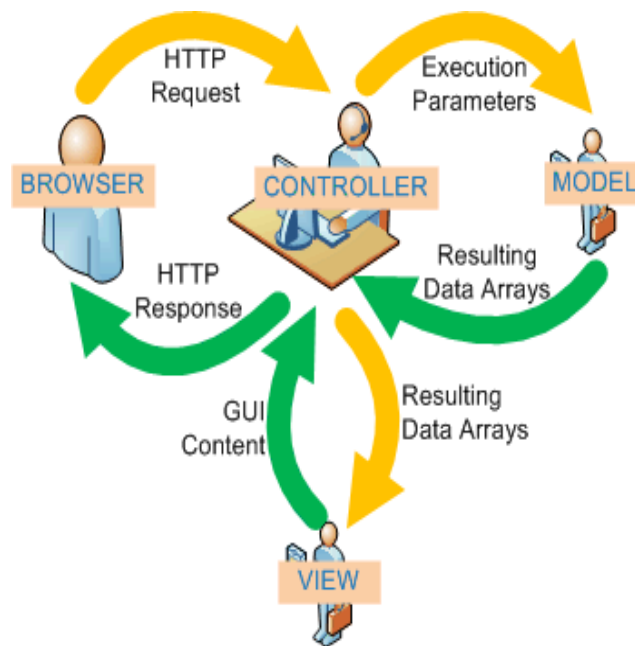
Framework adalah aplikasi *semi-complete* yang dapat digunakan untuk membuat aplikasi lain yang lebih spesifik (Husted 2003). *Framework* yang ideal merupakan intisari dari pendekatan terbaik dalam pengembangan perangkat lunak. Alasan pengembangan sebuah *framework* adalah penggunaan kembali perangkat lunak, jadi suatu perangkat lunak dikembangkan dari perangkat lunak yang telah ada.

Saat ini banyak sekali *framework* berbasis PHP, diantaranya yang paling populer adalah *Zend Framework*, *Symphony*, *Prado*, *CakePHP*, *CodeIgniter* dan lain-lain. Menurut beberapa survei, *CodeIgniter* adalah *framework* yang paling mudah dipelajari, karena kesederhanaan strukturnya dan kelengkapan dokumentasinya yang baik. Selain itu, kerapian kode (*clean and healthy code*) dan ukurannya yang ramping membuatnya lebih cepat diakses dibandingkan

dengan framework lainnya. Perlu diketahui nyaris (bahkan mungkin semua) framework menggunakan konsep yang sama dalam membangun suatu aplikasi, yaitu konsep MVC (*Model View Controller*). Kesimpulannya, kalau kita memahami MVC, maka secara garis besar kita sudah menguasai *framework*, apapun jenis frameworknya. Dan satu lagi kuncinya, pemahaman terhadap OOP (*Object Oriented Programming*) akan sangat membantu dalam menguasai *framework*.

2.1.4 Model View Controller (MVC)

Seperti kebanyakan *Software engineering*, konsep *Model-View-Controller* diciptakan oleh *programmer Smalltalk*. Lebih tepatnya diciptakan oleh salah seorang *programmer Smalltalk*, yaitu Trygve Reenskaug. Beliau memperkenalkan konsep ini pada paper yang beliau rilis tanggal 10 November 1979. Konsep ini sekarang populer dalam pemrograman lanjutan karena kemudahan dalam perawatan dan dapat digunakan oleh banyak programmer. MVC adalah sebuah metode dalam membuat aplikasi dengan memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol utama aplikasi (Nur widiyanto. 2010). Berikut adalah cara kerja MVC :



Gambar 2.1 cara kerja MVC

a. Model

Model bukan sebuah *Database* tetapi merupakan bagian (fungsi-fungsi) yang berhubungan langsung dengan *Database* untuk memanipulasi data seperti memasukkan data, pembaruan data, hapus data, dan lain-lain, namun tidak dapat berhubungan langsung dengan bagian *view*. Model merepresentasikan *knowledge* (pengetahuan). Model dapat berupa single object (jarang dipakai) atau terdiri dari objek-objek yang terstruktur dimana tiap objek memiliki fungsi tersendiri sesuai dengan kebutuhan. Pada framework PHP, model biasanya digunakan sebagai penghubung antara controller dengan *Database* untuk mengambil data pada *Database*. Hal ini merujuk pada konsep MVC dimana model digunakan sebagai representasi dari pengetahuan (*Database*). Model merupakan “kerangka” dari konten. Teks yang mengkomunikasikan informasi kepada *user*, berikut ini adalah contoh dari model :



Gambar 2.2 contoh Model

b. View

View, merupakan bagian yang mengatur tampilan ke pengguna. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada pengguna. Bagian ini tidak memiliki akses langsung terhadap bagian model. *View* merupakan representasi dari tampilan. Bagian dari konsep MVC ini digunakan sebagai pendukung Model dan Controller untuk menampilkan data yang *diretrieve* (ditampilkan) pada *Database*. CSS menambahkan tampilan visual pada file HTML. Kita dapat mengganti tampilan HTML cukup dengan mengganti CSS. CSS merupakan kulit dari sistem web. berikut ini adalah contoh dari view :



Gambar 2.3 contoh View

c. Controller

Controller merupakan penghubung antara user dengan sistem. *Controller* dapat menerima input dari user dan sekaligus dapat memberikan output yang dapat digunakan oleh user. *Controller* juga dapat menghubungkan antara *View* dengan *Model*. *Browser* bertanggung jawab dalam mentransformasikan dan menggabungkan baik HTML dan CSS kepada user agar dapat dinikmati dan dipakai sebagai informasi yang berguna. Kekuatan dan simplisitas dari sebuah website dengan konsep *Model – View – Controller* tidak dapat diragukan lagi. Dengan mempelajari konsep MVC, proses pembuatan aplikasi akan lebih terstruktur dan dapat dirawat dengan baik. berikut ini adalah contoh dari controller :



Gambar 2.4 contoh Controller

2.1.5 *Object Relational Mapping (ORM)*

Object Relational Mapping merupakan teknik otomasi dan transparansi dari *object persistence* ke dalam tabel pada *Database*, menggunakan metadata yang mendeskripsikan pemetaan antara objek dan *Database* (Bauer dan King 2007). ORM berperan dalam lapisan model dalam konsep MVC. Model adalah sebuah lapisan yang paling dekat dengan sumber data, baik itu berasal dari *Database*, *webservice*, maupun *file system*. *Object Relational Mapping (ORM)* adalah sebuah *framework* yang mengatasi perbedaan sistem *Database* yang bersifat relational dengan paradigma pengembangan aplikasi yang berorientasi objek. Selain itu, ORM juga menjembatani dialek SQL yang digunakan, sehingga apapun produk RDBMS yang digunakan tidak berpengaruh terhadap kode program. ORM merupakan solusi yang mengatasi perbedaan aspek-aspek ketidaksesuaian antara konsep pemrograman berorientasi objek dengan konsep *Database* relasional.

2.1.6 Pengertian *Database*

Database adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah di olah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya *Database* adalah media untuk menyimpan data agar dapat di akses dengan mudah dan cepat. Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan *Database* apapun bentuknya, entah berupa file teks ataupun *Database Management System* (Rosa, Shalahuddin, 2014). Kebutuhan *Database* dalam system informasi meliputi : memasukan, menyimpan, dan mengambil data, membuat laporan berdasarkan data yang telah disimpan. Tujuan dari dibuatnya tabel – tabel disini adalah untuk menyimpan data ke dalam tabel – tabel agar mudah di akses. Oleh karena itu, untuk merancang tabel yang akan di buat maka dibutuhkan pola pikir penyimpanan data nantinya jika dalam bentuk baris data (*record*) dimana setiap baris terdiri dari beberapa kolom .

2.1.7 DBMS (*Database Management System*)

Database Management System adalah suatu perangkat lunak yang ditujukan untuk menangani penciptaan, pemeliharaan dan pengendalian akses data (Abdul Kadir, 2009). Dengan menggunakan perangkat lunak ini pengelolaan data menjadi mudah dilakukan. Selain itu perangkat lunak ini juga menyediakan berbagai piranti yang berguna. Misalnya peranti yang memudahkan dalam membuat berbagai bentuk laporan. Sejauh ini banyak sekali produk DBMS yang beredar antara lain: *Oranle 9i*, *Microsoft SQL Server 7.0*, *Microsoft Access 200* merupakan contoh produk DBMS komersil yang terkenal di dunia. Produk non-komersil untuk pengelolaan *Database* juga tersedia dan dapat di unduh di

internet. MySQL ataupun PostgreSQL dapat menjadi pilihan untuk mengimplementasikan *Database*, walaupun fasilitas yang ditawarkan tidak selengkap produk komersial. Namun begitu bukan berarti *Database* yang didukung tidak handal. Berikut ini adalah komponen pada DBMS (Indrajani, 2014) :

- a. *Hardware* atau perangkat keras, diperlukan oleh DBMS dan aplikasi. Contohnya *personal computer, notebook, mainframe*, sampai sebuah jaringan computer.
- b. *Software* atau perangkat lunak, contohnya sistem operasi, *Software* untuk *Database*, *Software* pemrograman dan *Software* untuk mengatur jaringan.
- c. Data, merupakan komponen terpenting DMBS karena data merupakan penghubung antara komputer dengan manusia
- d. Prosedur, merupakan intruksi dan aturan yang menentukan perancangan dan penggunaan *Database*.
- e. Manusia, dimana peranan manusia sangatlah penting karena manusia yang menjalankan semua pekerjaan.

2.1.8 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah salah satu standar Bahasa yang banyak digunakan didunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan Bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks – teks pendukung (Rosa, Shalahuddin, 2014). UML hanya berfungsi untuk

melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataan UML paling banyak digunakan pada metodologi berorientasi objek. Seperti kita ketahui bahwa banyak hal di dunia sistem informasi yang tidak dapat dibakukan, semua tergantung pada kebutuhan, lingkungan dan konteksnya. Begitu juga dengan perkembangan penggunaan UML bergantung pada level abstraksi penggunaannya. Jadi belum tentu pandangan yang berbeda dalam penggunaan UML adalah suatu yang salah, tetapi perlu ditelaah dimanakah UML digunakan dan hal apa yang ingin divisualkan. Secara analogi jika dengan Bahasa sehari – hari, belum tentu menyampaikan bahasa dengan puisi adalah hal yang salah. Sistem informasi bukanlah ilmu pasti, maka jika ada banyak perbedaan merupakan hal yang wajar.

Konsep Pemodelan menggunakan UML, merupakan metode pemodelan berorientasi objek dan berbasis visual. Karenanya pemodelan menggunakan UML merupakan pemodelan objek yang fokus pada pendefinisian struktur statis dan model sistem informasi yang dinamis dari pada mendefinisikan data dan model proses yang tujuannya adalah pengembangan tradisional. UML menawarkan diagram yang dikelompokkan menjadi lima perspektif berbeda untuk memodelkan suatu sistem. Seperti satu set *blue print* yang digunakan untuk membangun sebuah rumah. Berikut ini adalah Diagram Dasar dalam UML :











1. Model *Usecase* Diagram

Usecase atau diagram *Usecase* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Usecase* mendeskripsikan sebuah interaksi satu atau lebih *actor* dengan sistem informasi yang akan dibuat. Secara kasar *Usecase* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi – fungsi itu (Rosa, Shalahuddin, 2014). Syarat penamaan *Usecase* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *Usecase* yaitu pendefinisian apa yang disebut *actor* dan *Usecase* .

- a. *Actor* merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari *actor* adalah gambar orang, tapi *actor* belum tentu orang.
- b. *Usecase* merupakan fungsionalitas yang disediakan sistem sebagai unit – unit yang saling bertukar pesan antar unit atau *actor*.

Usecase nantinya akan menjadi kelas proses pada diagram kelas sehingga perlu dipertimbangkan penamaan yang dilakukan apakah sudah layak menjadi kelas atau belum sesuai dengan aturan pendefinisian kelas yang baik. Setiap *usecase* dilengkapi dengan skenario. Skenario *usecase* adalah alur jalannya proses *Usecase* dari sisi *actor* dan sistem. Ada dua skenario *usecase*, yang pertama skenario normal adalah skenario bila sistem berjalan dengan normal tanpa terjadi kesalahan atau error. Yang kedua yaitu

scenario alternatif adalah *scenario* bila sistem tidak berjalan normal atau mengalami eror. *Scenario* normal dan alternatif dapat lebih dari satu. Alur dari *scenario* inilah yang nantinya menjadi dasar pembuatan diagram *sequence*. Diagram *Usecase* berguna dalam tiga hal : Menjelaskan fasilitas yang ada *requirement*, Komunikasi dengan klien, Membuat test dari kasus-kasus secara umum. Berikut ini adalah simbol *Usecase*

| SIMBOL | NAMA | KETERANGAN |
|-------------------------------------------------------------------------------------|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <i>Actor</i> | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> . |
|  | <i>Dependency</i> | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |
|  | <i>Generalization</i> | Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>). |
|  | <i>Include</i> | Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit. |
|  | <i>Extend</i> | Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan. |
|  | <i>Association</i> | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
|  | <i>System</i> | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
|  | <i>Use Case</i> | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor. |
|  | <i>Collaboration</i> | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>). |
|  | <i>Note</i> | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |

Gambar 2.5 Simbol *Usecase*

Diagram *usecase* mempunyai kelebihan dan kekurangan diantaranya sebagai berikut :

Kelebihan:

- a. Interaksi antara pengguna dan sistem lain dengan system yang akan di buat cukup tergambar dengan baik.

- b. Penggambaran dengan sederhana membuat identifikasi kebutuhan dengan *Usecase* dapat dengan lebih mudah untuk dipahami.
- c. Pendekatan identifikasi kebutuhan dapat berdasarkan *top down* (keinginan dari manajemen level atas) maupun *bottom up* (keinginan pengguna akhir).
- d. Dapat meng-include (memasukkan) fungsionalitas *Usecase* lain sebagai bagian dari proses dalam dirinya.
- e. dapat di-include oleh lebih dari satu *Usecase* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*.
- f. Dapat meng-extend (memperpanjang) *Usecase* lain dengan *behaviour*-nya sendiri.
- g. Sementara hubungan generalisasi antar *Usecase* menunjukkan bahwa *Usecase* yang satu merupakan spesialisasi dari yang lain.

Kelemahan:

- a. Kekurangan mengenai data masih kurang teridentifikasi dengan baik.

2. Diagram Struktur Statis

UML menawarkan dua diagram untuk memodelkan struktur statis sistem informasi, yaitu:

- a. **Class Diagram:** menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun

sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi (Rosa, Shalahuddin, 2014).

1. Atribut merupakan variabel – variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi – fungsi yang dimiliki oleh suatu kelas

Class diagram dibuat agar pembuat program membuat kelas – kelas sesuai rancangan didalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Kelas – kelas yang ada pada skruktur sistem harus dapat melakukan fungsi – fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak dapat membuat kelas – kelas didalam program sesuai dengan perancangan *class diagram*. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis – jenis kelas berikut :

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendefinisian *usecase* (*controller*)

Kelas yang menangani fungsi yang harus diambil dari pendefinisian usecase, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

4. Kelas yang diambil dari pendefinisian data (*model*)

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan di simpan ke *Database*. Semua tabel yang dibuat di *Database* dapat dijadikan kelas namun untuk tabel dari hasil relasi atau atribut multivalued pada erd yang dapat dijadikan kelas tersendiri .

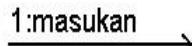
- b. **Object Diagram:** serupa dengan *class diagram*, tetapi *object diagram* memodelkan *instance object actual* dengan menunjukkan nilai-nilai saat ini dari atribut *instance*. *Object Diagram* menyajikan “*snapshot/potret*” tentang objek sistem pada point waktu tertentu. Diagram ini tidak digunakan sesering *class diagram*, tetapi saat digunakan dapat membantu seorang *developer* memahami struktur sistem secara lebih baik.

3. Diagram Interaksi

Diagram interaksi memodelkan sebuah interaksi, terdiri dari satu set objek, hubungan-hubungannya, dan pesan yang terkirim di antara objek.

Model diagram ini memodelkan *behavior* (kelakuan) sistem yang dinamis dan UML memiliki dua diagram untuk tujuan ini, yaitu:

- a. **Diagram rangkaian/Sequence Diagram:** menggambarkan kelakuan objek pada *Usecase* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek (Rosa, Shalahuddin, 2014). Oleh karena itu untuk menggambarkan diagram sequen maka harus diketahui objek – objek yang terlibat dalam sebuah *Usecase* beserta metode yang dimiliki kelas yang diinstansiasi menjadi objek. Membuat *diagram sequence* juga dibutuhkan untuk melihat *scenario* yang ada pada *usecase*. Banyak diagram sequen yang harus digambar adalah minimal sebanyak pendefinisian *usecase* yang dimiliki proses sendiri atau yang penting semua *usecase* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram *sequence* sehingga semakin banyak *usecase* didefinisikan maka diagram sequen yang harus dibuat juga semakin banyak. Berikut ini adalah simbol *sequen diagram*

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| aktor  atau nama_aktor | <ul style="list-style-type: none"> • orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi dan mendapat manfaat dari system. • Berpartisipasi secara berurutan dengan mengirimkan dan / atau menerima pesan. • Ditempatkan di bagian atas diagram. |
| objek objek:kelas | <p>Sebuah objek:</p> <ul style="list-style-type: none"> • Berpartisipasi secara berurutan dengan mengirimkan dan / atau menerima pesan. • Ditempatkan di bagian atas diagram. |
| Garis hidup objek  | <ul style="list-style-type: none"> • Menandakan kehidupan obyek selama urutan. • diakhiri tanda X pada titik di mana kelas tidak lagi berinteraksi. |
| Objek sedang aktif berinteraksi  | <p>Fokus kontrol:</p> <ul style="list-style-type: none"> • Adalah persegi panjang yang sempit panjang ditempatkan di atas sebuah garis hidup. • Menandakan ketika suatu objek mengirim atau menerima pesan. |
| pesan  | <p>objek mengirim satu pesan ke objek lain</p> |
|  | <p>menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p> |
|  | <p>menyatakan bahwa suatu objek mengirimkan masukan ke objek lainnya arah panah mengarah pada objek yang dikirim</p> |
|  | <p>objek/metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p> |
|  | <p>menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy</p> |

Gambar 2.6 Simbol *Sequence Diagram*

Penomoran pesan berdasarkan urutan interaksi pesan. Penggambaran letak pesan harus berurutan, pesan yang lebih atas dari lainnya adalah pesan yang berjalan terlebih dahulu.

Semua metode didalam kelas harus ada di dalam diagram *sequence*, jika tidak ada berarti perancangan metode didalam kelas itu kurang baik. Hal ini dikarenakan ada metode yang tidak dapat dipertanggungjawabkan kegunaanya.

2.1.9 Model – Model Proses Pengembangan Perangkat Lunak

Model proses untuk rekasya perangkat lunak dipilih berdasarkan sifat aplikasi dan proyeknya, metode dan alat – alat bantu yang akan dipakai dan control serta penyampaian yang dibutuhkan. Pada pengembangan aplikasi PSB menggunakan metode pengembangan *Waterfall*. Berikut ini adalah macam – macam model proses yang berbeda pada perangkat lunak.

2.1.9.1 Model Sekuensial Linier atau *Waterfall*

Sekuensial linier sering disebut juga dengan “siklus kehidupan klasik” atau “model air terjun. Sekuen linier mengusulkan sebuah pendekatan pada pengembang perangkat lunak yang sistematis dan sequential yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan (Roger Pressman, 2007). Model sekuensial linier adalah paradigma rekayasa perangkat lunak yang paling luas dipakai dan paling tua (Hanna, M , 1995). Dimodelkan setelah siklus rekayasa konvensional, model sekuensial linier melingkupi aktifitas sebagai berikut :

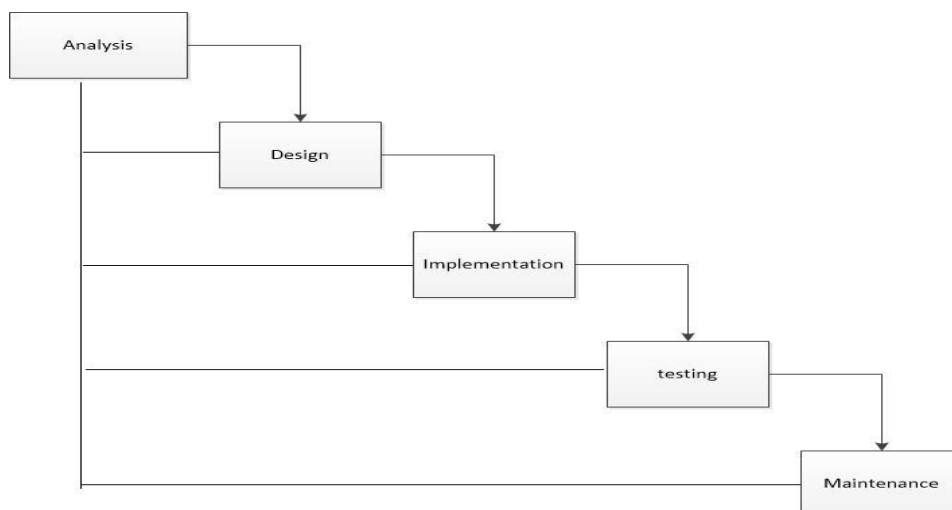
- a. Rekayasa dan Pemodelan sistem/informasi.** Karena perangkat lunak selalu merupakan bagian dari sebuah sistem yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan perangkat lunak

tersebut. Pandangan sistem ini penting ketika perangkat lunak harus berhubungan dengan elemen- elemen lain seperti perangkat lunak, manusia, dan *Database*. Rekayasa dan analisis menyangkut pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis.

- b. **Analisis Kebutuhan perangkat lunak.** Proses pengumpulan kebutuhan diintensifkan dan difokuskan khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, rekayasa perangkat lunak (analisis) harus memahami domain informasi, tingkah laku, unjuk kerja, dan antar muka (*interface*) yang diperlukan. Kebutuhan baik untuk sistem maupun perangkat lunak didokumentasikan dan dilihat lagi dengan pelanggan.
- c. **Desain.** Desain perangkat lunak sebenarnya proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda ; struktur data, arsitektur perangkat lunak, representasi *interface*, detail (algoritma) *procedural*. Proses desain menerjemahkan kebutuhan kedalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode.
- d. **Generasi kode.** Desain harus diterjemahkan kedalam bentuk mesin yang dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.

- e. **Pengujian.** Sekali kode dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional yaitu mengarahkan pengujian untuk menemukan kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil yang sesuai dengan kebutuhan.
- f. **Pemeliharaan.** Perangkat lunak akan mengalami perubahan setelah disampaikan pada pelanggan. Perubahan akan terjadi karena kesalahan – kesalahan yang ditentukan, karena perangkat lunak harus disesuaikan untuk mengakomodasi perubahan- perubahan didalam lingkungan eksternal atau karena pelanggan membutuhkan perkembangan fungsional.

Berikut ini adalah gambar dari model sequensial linier atau waterfall :



Gambar 2.7 Model Sequensial Linier atau Waterfall

Berikut ini adalah Kelebihan dan Kelemahan dari model pengembangan

Sekuensial Linear atau Waterfall :

Kelebihan model *sekuensial linear atau waterfall :*

- a. Mudah diaplikasikan
- b. Memberikan template tentang metode analisis, desain, pengkodean, pengujian, dan pemeliharaan.

Kelemahan model *sekuensial linear atau waterfall :*

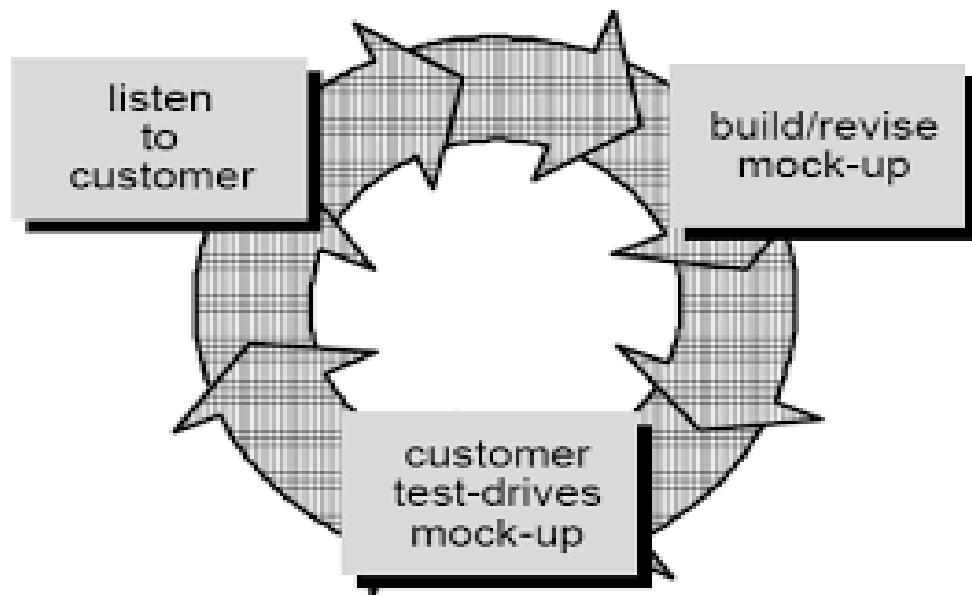
- a. Jarang sekali proyek riil mengikuti aliran sekuensial yang dianjurkan model karena model ini bisa melakukan iterasi tidak langsung. Hal ini berakibat ada perubahan yang diragukan pada saat proyek berjalan.
- b. Pelanggan sulit untuk menyatakan kebutuhan secara eksplisit sehingga sulit untuk mengakomodasi ketidak pastian pada saat awal proyek.
- c. Pelanggan harus bersikap sabar karena harus menunggu sampai akhir proyek dilalui. Sebuah kesalahan jika tidak diketahui dari awal akan menjadi masalah besar karena harus mengulang dari awal.
- d. Pengembang sering melakukan penundaan yang tidak perlu karena anggota tim proyek harus menunggu tim lain untuk melengkapi tugas karena memiliki ketergantungan hal ini menyebabkan penggunaan waktu tidak efisien.

2.1.9.2 Model Prototype

Prototype dimulai dari pengumpulan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Program *prototype* biasanya merupakan program yang belum jadi. Program ini biasanya menyediakan tampilan dengan

simulasi alur perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi (Rosa, Shalahuddin, 2014). Program *prototype* ini dievaluasi oleh pelanggan atau user sampai ditemukan spesifikasi sesuai dengan keinginan user.

Berikut ini adalah gambar dari model *prototype* :



Gambar 2.8 Model Prototype

Berikut ini adalah Kelebihan dan Kelemahan dari metode pengembangan

Prototype :

Kelebihan *prototype* adalah :

- a. Adanya komunikasi yang baik antara pengembang dan pelanggan.
- b. Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan
- c. Pelanggan berperan aktif dalam pengembangan sistem
- d. Lebih menghemat waktu dalam pengembangan sistem
- e. Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya.

Kelemahan *prototype* adalah :

- a. Pelanggan kadang tidak melihat atau menyadari bahwa perangkat lunak yang ada belum mencantumkan kualitas perangkat lunak secara keseluruhan dan juga belum memikirkan kemampuan pemeliharaan untuk jangka waktu lama.
- b. Pengembang biasanya ingin cepat menyelesaikan proyek. Sehingga, menggunakan algoritma dan bahasa pemrograman yang sederhana untuk membuat *protoptype* lebih cepat selesai tanpa memikirkan lebih lanjut bahwa program tersebut hanya merupakan cetak biru sistem.
- c. Hubungan pelanggan dengan *computer* yang disediakan mungkin tidak mencerminkan teknik perancangan yang baik.

Prototype bekerja dengan baik pada penerapan-penerapan yang bercirikan sebagai berikut :

- a. Resiko tinggi yaitu untuk masalah-masalah yang tidak terstruktur dengan baik, ada perubahan yang besar dari waktu ke waktu, dan adanya persyaratan data yang tidak menentu
- b. Interaksi pemakai penting. Sistem harus menyediakan dialog *online* antara pelanggan dan komputer.
- c. Perlunya penyelesaian yang cepat
- d. Perilaku pemakai yang sulit ditebak
- e. Sistem yang inovatif. Sistem tersebut membutuhkan cara penyelesaian masalah dan penggunaan perangkat keras yang mutakhir.
- f. Perkiraan tahap penggunaan sistem yang pendek.

2.1.9.3 Model Rapid Application Development (RAD)

Model *Rapid Application Development* (RAD) adalah sebuah model proses pengembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek (Roger Pressman, 2001). Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier dimana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen. Jika kebutuhan dipahami dengan baik proses RAD memungkinkan tim pengembang menciptakan “sistem fungsional yang utuh” dalam periode waktu yang sangat pendek kira – kira 60 sampai 90 hari (martin, 1994). Karena dipakai terutama pada aplikasi sistem konstruksi pendekatan RAD melingkupi fase sebagai berikut

a. *Bussines modeling*

Aliran infirmasi diantara fungsi – fungsi bisnis dimodelkan dengan suatu cara untuk menjawab pertanyaan berikut : informasi apa yang mengendalikann proses bisnis? Informasi apa yang dimunculkan? Kemana informasi itu pergi? Siapa yang memproses?

b. *Data modeling*

Aliran informasi yang didefinisikan sebagai sabagian dari fase bussness modeling disaring kedalam serangkaian objek data yang dibutuhkan untuk menopang bisnis tersebut.

c. *Proses modeling*

Aliran informasi didefinisikan didalam fase data modeling ditransformasikan untuk mencapai aliran informasi yang perlu dibagi implementasi sebuah fungsi bisnis.gambaran pemrosesan diciptakan

untuk menambah, memodifikasi, menghapus, dan mendapatkan kembali sebuah objek data.

d. Application generation

RAD mengasumsikan pemakaian teknik generasi ke empat. Selain menciptakan perangkat lunak dengan menggunakan Bahasa pemrograman generasi ketiga yang konvensional, RAD lebih banyak memproses kerja untuk memakai lagi komponen program yang ada atau menciptakan komponen yang bisa dipakai untuk memfasilitasi konstruksi perangkat lunak.

e. Testing and turnover

Karena proses RAD menekankan pada pemakaian kembali, banyak komponen program telah di uji. Hal ini mengurangi keseluruhan waktu pengujian. Tetapi komponen baru harus di ujikan semua interface harus dilatih secara penuh.

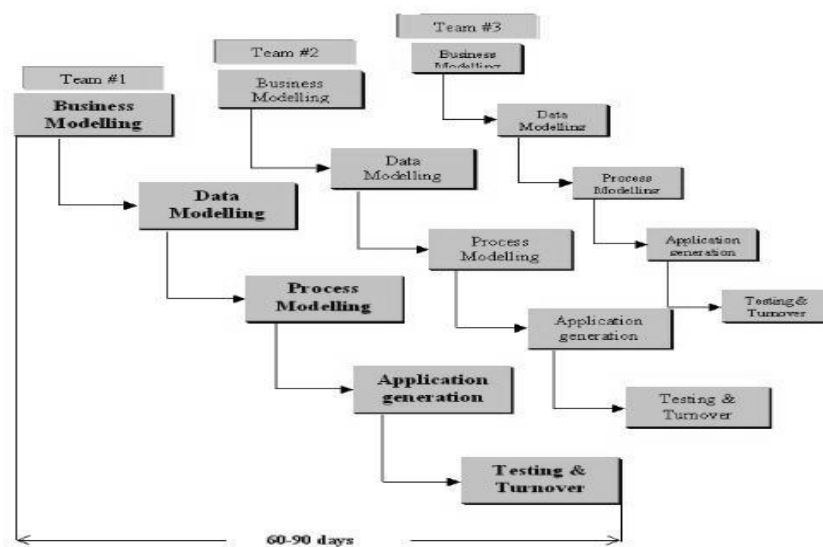


Figure 1.5 – RAD Model

Gambar 2.9 Model RAD

Berikut ini adalah kelebihan dan Kelemahan Model RAD :

Kelebihan Model RAD adalah :

- a. Setiap fungsi mayor dapat dimodulkan dalam waktu tertentu kurang dari 3 bulan dan dapat dibicarakan oleh tim RAD yang terpisah dan kemudian diintegrasikan sehingga waktunya lebih efisien.
- b. RAD mengikuti tahapan pengembangan sistem seperti umumnya, tetapi mempunyai kemampuan untuk menggunakan kembali komponen yang ada (reuseable object) sehingga pengembang pengembang tidak perlu membuat dari awal lagi dan waktu lebih lanjut.

Kelemahan Model RAD adalah :

1. Proyek yang besar dan berskala, RAD memerlukan daya manusia yang memadai untuk menciptakan jumlah tim yang baik.
2. RAD menuntut pengembang dan pelanggan memiliki komitmen dalam aktivitas rapid fire yang diperlukan untuk melengkapi sebuah sistem dalam waktu yang singkat. Jika komitmen tersebut tidak ada maka proyek RAD akan gagal.

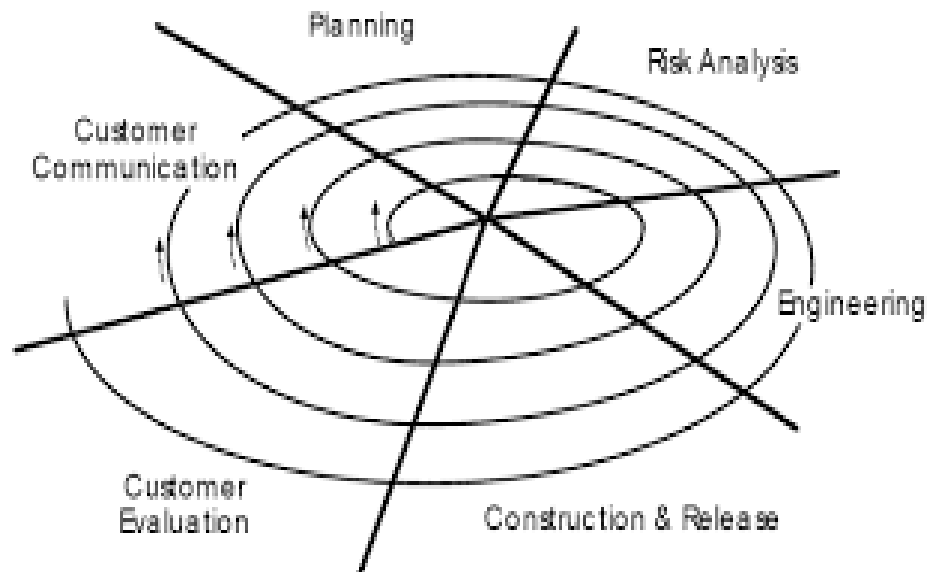
2.1.9.4 Model Spiral

Model spiral memasangkan iteratif pada model *prototype* dengan control dan aspek sistematis yang diambil dari model air terjun. Model spiral menyediakan pengembangan dengan cara cepat dengan perangkat lunak yang memiliki versi yang terus bertambah fungsinya (*increment*) (Rosa, Shalahuddin, 2014). Pada iterasi awal maka yang dihasilkan adalah *prototype* sedangkan pada

iterasi akhir yang dihasilkan adalah perangkat lunak yang sudah lengkap. model spiral dibagi dari beberapa kerangka aktifitas sebagai berikut :

- a. Komunikasi dengan pelanggan (*customer communication*)
Aktifitas yang diperlukan untuk membangun komunikasi yang efektif antara pengembang dan pelanggan
- b. Perencanaan (*planning*)
Aktifitas yang diperlukan untuk mendefinisikan sumberdaya, waktu , dan informasi terkait proyek
- c. Analisis resiko (*risk analysis*)
Aktifitas ini diperlukan untuk memperkirakan resiko dari segi teknis maupun manajemen
- d. Rekayasa (*engineering*)
Aktifitas ini diperlukan untuk membangun satu atau lebih representasi dari aplikasi perangkat lunak
- e. Kontruksi (*contrutionadn release*)
Aktifitas ini dibutuhkan untuk mengkrontuksi, menguji, melakukan instalasi, dan menyediakan dukungan terhadap user.
- f. Evaluasi pelanggan (*customer evaluation*)
Aktifitas ini dibutuhkan untuk mendapatkan umpan balik berdasarkan evaluasi represents perangkat lunak yang dihasilkan dari proses rekayasa dan implementasi pada tahap instalasi.

Berikut ini adalah gambar dari model spiral :



Gambar 2.10 Ilustrasi Model Spiral

Berikut ini adalah kelebihan dan kelemahan dari model *spiral*

Kelebihan model *Spiral* adalah :

- a. Dapat disesuaikan agar perangkat lunak bisa dipakai selama hidup perangkat lunak *computer*.
- b. Lebih cocok untuk pengembangan sistem dan perangkat lunak skala besar.
- c. Pengembang dan pengguna dapat lebih mudah memahami dan bereaksi terhadap resiko setiap tingkat evolusi karena perangkat lunak terus bekerja selama proses.
- d. Menggunakan *prototype* sebagai mekanisme pengurangan resiko dan pada setiap keadaan di dalam evolusi produk.
- e. Tetap mengikuti langkah-langkah dalam siklus kehidupan klasik dan memasukkannya ke dalam kerangka kerja *iterative*.
- f. Membutuhkan pertimbangan langsung terhadap resiko teknis sehingga mengurangi resiko sebelum menjadi permasalahan yang serius.

Kelemahan model Spriral adalah :

- a. Sulit untuk meyakinkan pelanggan bahwa pendekatan *evolusioner* ini bisa dikontrol.
- b. Memerlukan penaksiran resiko yang masuk akal dan menjadi masalah yang serius jika resiko mayor tidak ditemukan dan diatur.
- c. Butuh waktu lama untuk menerapkan paradigma ini menjadi kepastian yang absolut.

2.1.10 Pengujian

Pengujian merupakan bagian yang tidak terpisahkan dari perangkat lunak. Proses pengujian juga mempengaruhi masa penggunaan perangkat lunak. Semakin terperinci proses pengujian yang dilakukan, semakin lama rentang waktu yang akan diperlukan pada satu pemeliharaan dan untuk proses selanjutnya. Pentingnya pengujian perangkat lunak dan implikasinya mengacu pada kualitas perangkat lunak yang tidak dapat terlalu ditekan karena melibatkan sederetan aktivitas produksi dengan peluang terjadinya kesalahan manusia yang sangat besar dan ketidakmampuan manusia untuk melakukan dan berkomunikasi dengan sempurna. Sejumlah aturan yang berfungsi sebagai sasaran pengujian pada perangkat lunak adalah sebagai berikut :

- a. Pengujian adalah proses eksekusi suatu program untuk menemukan kesalahan
- b. Kasus pengujian yang baik adalah kasus pengujian yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya

- c. Pengujian yang sukses adalah pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya

2.1.10.1 Black Box Testing

Black box testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa dibalik bungkus hitamnya. Berikut ini adalah jenis – jenis *Black box testing* :

a. Pengujian Fungsional (*functional testing*)

Perangkat lunak diuji untuk persyaratan fungsional. Pengujian dilakukan dalam bentuk tertulis untuk memeriksa apakah aplikasi berjalan seperti yang diharapkan. Walaupun pengujian fungsional sering dilakukan dibagian akhir dari siklus pengembangan, masing – masing komponen dan proses dapat diuji pada awal pengembangan, bahkan sebelum sistem berfungsi, pengujian ini sudah dapat dilakukan pada seluruh sistem. Pengujian fungsional meliputi seberapa baik sistem melaksanakan fungsinya, termasuk perintah – perintah pengguna, manipulasi data, pencarian dan proses bisnis, pengguna layar, dan integrasi. Pengujian fungsional juga meliputi permukaan yang jelas dari jenis fungsi – fungsi, serta operasi *back-end* seperti, keamanan dan bagaimana meningkatkan sistem (Janner Simartama, 2010).

b. Pengujian Tegangan (*stress testing*)

Pengujian tegangan berkaitan dengan kualitas aplikasi di dalam lingkungan. Idennya adalah untuk menciptakan sebuah lingkungan yang lebih menuntut aplikasi, tidak seperti saat aplikasi dijalankan pada beban kerja normal. Pengujian ini adalah hal yang paling sulit, cukup kompleks dilakukan dan memerlukan upaya bersama oleh semua tim (Janner Simartama, 2010).

c. Pengujian Beban (*load testing*)

Pada pengujian beban, aplikasi akan diuji dengan beban berat atau masukan, seperti yang terjadi pada pengujian situs web untuk mengetahui apakah aplikasi/situs gagal atau kinerjanya menurun. Pengujian beban beroperasi pada tingkat beban standar biasanya beban tertinggi akan diberikan ketika sistem dapat menerima dan tetap berfungsi dengan baik. Perlu diketahui bahwa pengujian beban tidak bertujuan untuk merusak sistem dengan banyak hal, namun mencoba untuk menjaga agar sistem selalu kuat berjalan dengan lancar (Janner Simartama, 2010).

d. Pengujian khusus (*ad-hoc testing*)

Jenis pengujian ini dilakukan tanpa penciptaan rencana pengujian (*test plan*) atau kasus pengujian (*test case*). Pengujian khusus membantu dalam menentukan lingkup dan durasi dari berbagai pengujian lainnya dan juga membantu para penguji dalam mempelajari aplikasi sebelum memulai pengujian dengan pengujian lainnya. Pengujian ini merupakan metode pengujian formal yang paling sedikit. Salah satu

pengguna terbaik dari pengujian khusus adalah untuk penemuan. Membaca persyaratan atau spesifikasi (jika ada) jarang memberikan panduan yang jelas mengenai bagaimana sebuah program benar-benar bertindak bahkan dokumentasi pengguna tidak mengungkap “*look and fell*” dari sebuah program. Pengujian khusus dapat menemukan lubang dalam pengujian strategi dan dapat mengekspos hubungan di antara subsistem lain yang tidak jelas. Dengan cara ini, pengujian khusus berfungsi sebagai alat untuk memeriksa kelengkapan yang di uji (Janner Simartama, 2010).

e. Pengujian penyelidikan (*exploratory testing*)

Pengujian penyelidikan mirip dengan pengujian khusus dan dilakukan untuk mempelajari/mencari aplikasi. Pengujian penyelidikan perangkat lunak ini merupakan pendekatan yang menyenangkan untuk pengujian (Janner Simartama, 2010).

f. Pengujian usabilitas (*usability testing*)

Pengujian ini disebut juga sebagai pengujian untuk keakraban pengguna (*testing for user friendliness*). Pengujian ini dilakukan jika antarmuka pengguna dari aplikasinya penting dan harus spesifik untuk jenis pengguna tertentu. Pengujian usabilitas adalah proses yang bekerja dengan pengguna akhir secara langsung maupun tidak langsung untuk menilai bagaimana pengguna merasakan paket perangkat lunak dan bagaimana mereka berinteraksi dengannya. Proses ini akan membongkar area kesulitan pengguna seperti halnya area kekuatan. Tujuan dari pengujian usabilitas harus membatasi dan

menghilangkan kesulitan bagi pengguna dan untuk mempengaruhi area yang kuat untuk usability maksimum. Pengujian ini idealnya melibatkan masukan dari pengguna secara langsung maupun tidak langsung (mengamati perilaku) dan bila memungkinkan melibatkan komputer yang didukung umpan balik (Janner Simartama, 2010).

g. Pengujian pemulihan (*recovery testing*)

Pengujian pemulihan (*recovery testing*) pada dasarnya dilakukan untuk memeriksa seberapa cepat dan baiknya aplikasi bisa pulih terhadap semua jenis *crash* atau kegagalan *Hardware*, masalah benacana, dan lain-lain. Jenis atau taraf pemulihan ditetapkan dalam persyaratan spesifikasi (Janner Simartama, 2010).

h. Pengujian *scenario* (*scenario testing*)

Pengujian *scenario* adalah pengujian yang realistis kredibel dan memotivasi stakeholder, tantangan untuk program dan mempermudah penguji untuk melakukan evaluasi. Pengujian ini menyediakan kombinasi variable dan fungsi yang sangat berarti daripada kombinasi buatan yang anda dapatkan dengan pengujian domain atau desain pengujian kombinasi (Janner Simartama, 2010).

i. Pengujian regresi (*regression testing*)

Pengujian regresi adalah gaya pengujian yang berfokus pada pengujian ulang (*retesting*) setelah ada perubahan. Pada pengujian regresi berorientasi resiko (*risk-oriented regression testing*) daerah yang sama yang telah uji, akan kita uji lagi dengan pengujian yang berbeda (semakin kompleks). Usaha pengujian regresi bertujuan untuk

mengurangi resiko diantaranya perubahan yang dimaksudkan untuk memperbaiki *bug* yang gagal, dan beberapa perubahan memiliki efek samping, tidak memperbaiki *bug* lama atau memperkenalkan *bug* baru (Janner Simartama, 2010).

j. Pengujian penerimaan pengguna (*user acceptance*)

Pada jenis pengujian ini, perangkat lunak akan diserahkan kepada pengguna untuk mengetahui apakah perangkat lunak memenuhi harapan pengguna dan bekerja seperti yang diharapkan. Pada pengembangan perangkat lunak, *user acceptance testing* (UAT), juga disebut pengujian beta (*beta testing*), pengujian aplikasi (*application testing*) dan pengujian penggunaan akhir (*end user testing*) adalah tahapan pengembangan perangkat lunak ketika perangkat lunak diuji pada “dunia nyata” yang dimaksud oleh pengguna . UAT dapat dilakukan dengan *in-house testing* dengan membayar relawan atau subjek pengujian menggunakan perangkat lunak atau biasanya mendistribusikan perangkat lunak secara luas dengan melakukan pengujian versi yang tersedia secara gratis untuk diunduh melalui web. Pengalaman awal pengguna akan diteruskan kembali kepada para pengembang yang membuat perubahan sebelum akhirnya melepas perangkat lunak komersial (Janner Simartama, 2010).

k. Pengujian Alfa (*alpha testing*)

Pada jenis pengujian ini, pengguna akan diundang ke pusat pengembangan. Pengguna akan menggunakan aplikasi dan pengembang mencatat setiap masukan atau tindakan yang dilakukan

oleh pengguna. Semua jenis perilaku yang tidak normal dari sistem dicatat dan dikoreksi oleh para pengembang (Janner Simartama, 2010).

1. Pengujian Beta (*beta testing*)

Pada jenis pengujian ini, perangkat lunak didistribusikan sebagai sebuah versi beta dengan pengguna yang menguji aplikasi. *Bug* yang terjadi akan dilaporkan kepada pengembang. Pengujian beta dilakukan setelah pengujian alfa. Versi perangkat lunak yang terkenal dengan sebutan versi beta dirilis untuk pengguna yang terbatas diluar perusahaan. Perangkat lunak dilepas ke kelompok masyarakat agar dapat memastikan bahwa perangkat lunak tersebut memiliki beberapa kesalahan atau *bug* (Janner Simartama, 2010).

2.1.10.2 White Box Testing

White box testing adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara sekilas dapat diambil kesimpulan white box testing merupakan petunjuk untuk mendapatkan program yang benar secara 100%. Berikut ini adalah jenis – jenis *White box testing* :

a. Pengujian unit (*unit testing*)

Pengembang melaksanakan pengujian unit untuk memeriksa apakah modul atau kode unit bekerja dengan baik. Pengujian unit berada pada tingkat yang sangat dasar seperti ketika unit kode dikembangkan atau fungsi tertentu dibangun. Pengujian unit

berkaitan dengan unit secara keseluruhan. Lingkup yang tepat dari unit ditinggalkan kepada interpretasi, pendukung kode pengujian, kadang disebut perancah (*scaffolding*), mungkin diperlukan untuk mendukung setiap pengujian. Focus ini juga disebut *black box testing* karena hanya perincian antarmuka yang terlihat untuk pengujian. Jenis pengujian digerakan oleh tim arsitektur dan implementasi. Focus ini juga disebut *black box testing* karena hanya perincian antarmuka yang terlihat untuk pengujian. (Janner Simartama, 2010).

b. Analisis statis dan dinamis (*static and dynamic analysis*)

Analisis statis dilibatkan melalui kode untuk mengetahui segala kemungkinan cacat dalam kode, sedangkan analisis dinamis akan melibatkan pelaksanaan kode dan menganalisis hasilnya (Janner Simartama, 2010).

c. Pengujian mutase (*mutation testing*)

Pada pengujian ini aplikasi diuji untuk kode yang telah di modifikasi setelah pemasangan *bug* / cacat tertentu. Hal ini juga membantu dalam menemukan kode dan strategi pengkodean yang dapat membantu dalam mengembangkan fungsi secara efektif (Janner Simartama, 2010).

d. Cakupan pernyataan (*statement coverage*)

Dalam hal ini jenis pengujian kode dijalankan dengan setiap pernyataan dari aplikasi yang dijalankan minimal sekali. Hal

tersebut membantu dalam memastikan semua pernyataan untuk dijalankan tanpa efek samping (Janner Simartama, 2010).

e. Cakupan cabang (*branch coverage*)

Tidak ada aplikasi perangkat lunak yang dapat ditulis dalam cara pengkodean, di beberapa titik kita perlu mengetahui cakupan cabang untuk melakukan fungsi tertentu. Pengujian cakupan cabang membantu memvalidasi semua cabang didalam kode dan memastikan bahwa tidak ada yang mengarah ke percabangan perilaku abnormal dari aplikasi (Janner Simartama, 2010).

2.1.11. PSB

PSB adalah proses pendaftaran, penyeleksian, siswa dari sekolah lama untuk menjadi siswa baru di satu sekolah dengan beberapa persyaratan yang telah ditentukan oleh sekolah. PSB merupakan salah satu kewajiban pihak sekolah dan Dinas Pendidikan setiap tahun ajaran baru. PSB merupakan suatu proses administrasi yang terjadi setiap tahun untuk seleksi calon siswa berdasarkan nilai akademik agar dapat melanjutkan pendidikan pada jenjang yang lebih tinggi. Calon siswa yang dimaksud adalah siswa baru yang akan mendaftar pada jenjang SMP atau SMA/SMK negeri. Apabila setiap tahunnya proses PSB selalu dilakukan secara manual pada masing-masing sub rayon, maka akan menyebabkan berbagai kesulitan dari beberapa pihak yang terkait.

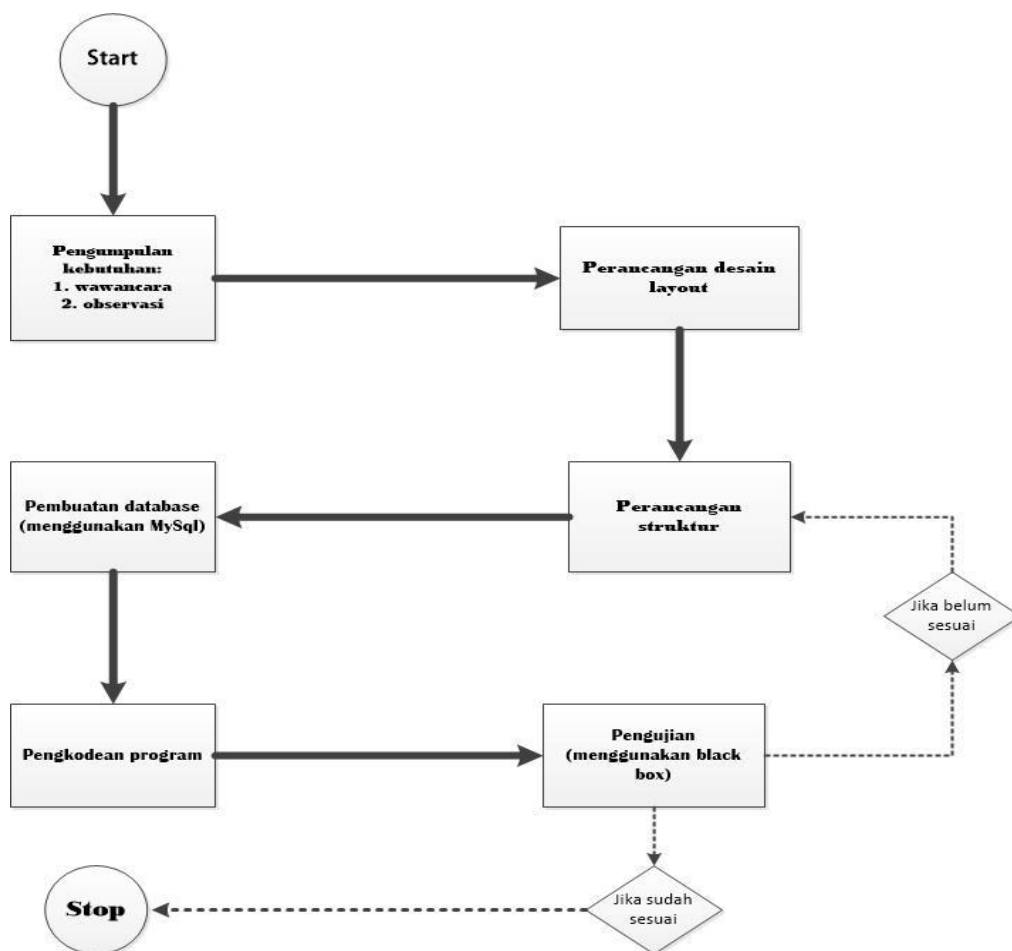
Pendaftaran adalah proses, cara, perbuatan mendaftar (mendaftarkan); pencatatan nama, alamat, dan sebagainya ke dalam daftar (Kamus Besar Bahasa Indonesia, 2008:285). Siswa adalah anggota masyarakat yang berusaha

mengembangkan dirinya melalui proses pendidikan pada jalur jenjang dan jenis pendidikan tertentu (Mendiknas,2003).

2.2. Kerangka Berpikir

Perkembangan teknologi yang sangat pesat saat ini, khususnya perkembangan teknologi informasi dan komunikasi menyebabkan setiap orang lebih mudah dan cepat mendapatkan informasi . teknologi tersebut juga dapat diterapkan didalam dunia pendidikan. YPI Assa'adah Pondok Kelapamerupakan salah satu sekolah swasta unggulan di pondok kelapa sehingga menjadikan sekolah ini banyak dijadikan pilihan oleh calon siswa dan orang tua murid. Setiap tahunnya YPI As Sa'adah Pondok Kelapa menerima siswa baru kurang lebih 3 sampai 4 kelas. Akan tetapi sampai saat ini masih menggunakan sistem informasi yang belum terkomputerisasi atau manual. Setiap calon siswa harus mengisi formulir pendaftaran diselembar kertas, dan panitia mencatat setiap data calon siswa kedalam buku besar dan menyimpan data dalam sebuah ruangan yang berisi dokumen – dokumen lain sehingga sering menimbulkan lambatnya suatu informasi yang dihasilkan atau harus ditingkatkan lagi terutama dalam bidang pendaftaran, penyeleksian dan pengelompokan kelas siswa baru di YPI As Sa'adah Pondok Kelapa. PSB merupakan kegiatan yang amat penting karena proses tersebut mempunyai nilai strategis guna menjaring calon siswa yang berkualitas.. Sistem informasi PSB di YPI As Sa'adah Pondok Kelapayang ditujukan untuk penyediaan informasi tentang data siswa yang akan di terima. Dengan komputer sebagai alat bantunya dalam menyelesaikan masalah yang

dihadapi oleh sekolah dan sudah waktunya memakai fasilitas komputer yang meringankan sedikit beban pekerjaan dan diharapkan dapat membantu YPI As Sa'adah Pondok Kelapauntuk mencapai tujuan dalam waktu yang singkat. Adapun proses yang digunakan dalam pembuatan sistem ini adalah dengan menggunakan PHP dan SQL. Sehingga dalam proses PSB lebih mudah. Gambar 2.6 merupakan bagan kerangka berpikir.



Gambar 2.11 Kerangka Berpikir

BAB III

METODOLOGI PENELITIAN

3.1 Tujuan Operasional Penelitian

Tujuan dari penelitian ini adalah mengembangkan sebuah sistem informasi PSB yang dapat mempermudah untuk melakukan pendaftaran dengan menggunakan bahasa pemrograman PHP dan basis data SQL pada pengembangan aplikasi PSB di YPI As Sa'adah Pondok Kelapa , Jaktim.

3.2 Tempat dan Waktu Penelitian

Penelitian dilakukan di Laboratorium Multimedia Jurusan Teknik Elektro Universitas Negeri Jakarta sejak bulan April 2015 hingga september 2015.

3.3 Metode Penelitian

Dalam memecahkan permasalahan diperlukan strategi pengembangan yang terdiri dari proses, metode dan alat perancangan. Oleh karena itu dalam pembuatan sistem informasi PSB ini menggunakan metode pengembangan sistem waterfall untuk pengembangan perangkat lunak dan eksperimen untuk metode penelitian

3.3.1 Tahapan Penelitian

Tahap-tahap penelitian yang digunakan pada pengembangan aplikasi PSB ini adalah:

a. Observasi

Observasi dilakukan untuk mendapatkan informasi-informasi yang dibutuhkan untuk melanjutkan suatu penelitian. Peneliti terjun langsung ke sekolah untuk mengetahui lebih detailnya mengenai proses PSB. Hasil observasi akan menjadi pedoman untuk mengembangkan aplikasi PSB.

b. Wawancara

Wawancara dilakukan dengan cara melakukan berdiskusi dengan pihak yang bisa memberikan informasi mengenai aplikasi yang penulis buat. Peneliti melakukan wawancara secara langsung dengan beberapa pihak yang pertama kepada pihak yayasan yang kedua kepada pegawai tata usaha dan yang ketiga kepada kepala sekolah. Hal ini dilakukan untuk mendapatkan data-data spesifik mengenai proses PSB pada YPI As Sa'adah Pondok Kelapa seperti cara pendaftaran, formulir dan bentuk laporan yang terkait. Hasil wawancara terlampir.

c. Data Penelitian

Data mengenai pendaftaran siswa baru pada tahun 2015 – 2016 yang ada di YPI As Sa'adah Pondok Kelapa.

3.4 Alat dan Bahan Penelitian

Berikut ini adalah perangkat yang digunakan selama penelitian:

a. Perangkat Keras

- 1) Komputer dengan spesifikasi Processor intel(R) core™ I5-3340M
CPU @2.70 GHz, RAM 4.00 GB, Hardisk 500GB
- 2) Monitor 14"
- 3) printer

b. Perangkat Lunak

- 1) Sistem Operasi Windows 8
- 2) XAMPP versi 1.8.3
- 3) Web Browser Mozilla Firefox, chrome
- 4) Notepad ++
- 5) Relational rose

3.5 Langkah – langkah Penelitian.

Langkah – langkah yang di lakukan pada penelitian ini mengikuti metode yang dipakai dalam pengembangan perangkat lunak. Pengembangan aplikasi PSB menggunakan Metode *Waterfall* atau sering disebut model sekuensial linier. Berikut langkah-langkah penelitian yang dilakukan :

3.5.1 Analisis

Analisis bertujuan untuk menentukan hal-hal rinci yang akan dikerjakan oleh perangkat lunak. Analisis mencakup studi kelayakan dan analisis kebutuhan. Pengembangan aplikasi PSB bertujuan untuk melakukan registrasi atau pendaftaran siswa baru. pada tahap penyeleksian PSB ini untuk jenjang TK dan SD menggunakan batas umur sesuai dengan peraturan yang berlaku di kementrian pendidikan dan SMP dan SMK menggunakan NEM atau nilai SKHUN sesuai dengan standar sekolah. Tidak semua data siswa Yayasan Perguruan Islam Assa'adah yang akan digunakan dalam proses pendaftaran siswa baru, hanya data siswa baru tahun ajaran 2015-2016 dari jenjang TK, SD, SMP, dan SMK yang ada di YPI As Sa'adah Pondok Kelapa .

Setelah melakukan studi kelayakan, kemudian analisis kebutuhan dilakukan dengan observasi lapangan dan wawancara. Berikut adalah beberapa analisis kebutuhan yang akan membentuk spesifikasi kebutuhan, dan beberapa diantaranya akan direalisasikan pada penelitian ini:

1. Analisis kebutuhan Admin :

- a. Admin dapat membuat akun dengan mengisi data diri
- b. Admin dapat melihat, mengupdate dan menghapus data calon siswa
- c. Admin dapat mengatur kuota dan nilai minimal untuk seleksi
- d. Admin dapat mengatur halaman depan berupa informasi terbaru
- e. Admin dapat mengexport data siswa yang diterima kedalam bentuk csv atau excel.
- f. Admin dapat mengatur pembukaan dan penutupan pendaftaran.
- g. Admin dapat melihat grafik pendaftaran yang sudah masuk.

2. Analisis kebutuhan user atau calon siswa :

- a. *User* dapat melihat informasi mengenai pendaftaran
- b. *User* membuat akun sebelum mendaftar
- c. *User login* dengan akun yang dibuat
- d. *User* dapat mengisi form pendaftaran dengan data diri yang benar
- e. *User* dapat mencetak formulir pembayaran PSB
- f. *User* dapat mencetak biodata atau bukti pendaftaran
- g. *User* dapat melihat pengumuman hasil seleksi

Berdasarkan kebutuhan diatas, maka dibuatlah pengembangan aplikasi PSB berdasarkan kebutuhan yang ada. Tugas utama dari aplikasi PSB ini adalah seleksi calon siswa jenjang TK dan SD menggunakan batas umur sesuai dengan peraturan yang berlaku di kementerian pendidikan dan SMP dan SMK menggunakan NEM atau nilai SKHUN sesuai dengan standar sekolah. Setelah seleksi dilakukan hasil akhir akan dimanfaatkan untuk menentukan calon siswa diterima atau tidak.

Aplikasi PSB akan dikelola oleh admin yang bertugas mengelola data calon siswa diantaranya mengubah, menghapus, menambah, melihat dan mengolah laporan.

3.5.2 Desain

Tahap desain pada penelitian ini adalah membuat desain rancangan halaman – halaman yang diperlukan dalam aplikasi PSB ini. Setelah itu membuat rancangan dokumentasi sistem dengan menggunakan *Unified Modeling Language* (UML) yaitu dengan dibuatnya *Usecase* admin dan *user*. *Usecase* mendeskripsikan sebuah interaksi satu atau lebih *actor* dengan sistem informasi yang akan dibuat. Pada pengembangan aplikasi PSB ada dua *Usecase* yaitu *Usecase* admin dan *Usecase* *user*. Selanjutnya perancangan *sequence diagram* yang menggambarkan kelakuan objek pada *Usecase* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Membuat diagram *sequence* juga dibutuhkan untuk melihat *scenario* yang ada pada *Usecase*. Pada sistem ini ada 13 *sequence diagram*. Dan yang terakhir adalah *class diagram* yang menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun sistem. *Class diagram* dibuat agar pembuat program membuat kelas – kelas sesuai rancangan didalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Pada pengembangan aplikasi PSB ini terdiri dari empat belas *sequence diagram* sebagai berikut : delapan *sequence diagram* admin, dan enam *sequence diagram* *user*. Untuk lebih detailnya akan dibahas di bab IV.

3.5.3 *Implementation*

Setelah desain dilakukan, tahap selanjutnya adalah membuat kode program. Coding yang dilakukan penulis dalam *management* data yang diolah dalam satu paket aplikasi yang tergabung, yaitu XAMPP versi 1.8.3, dimana didalamnya menggunakan PHP.5.5.15 sebagai bahasa pemrograman dan mysql versi 5.0.41 sebagai data basenya, sedangkan webservernya menggunakan Apache/2.2.4

3.5.4 **Pengujian.**

Pengujian dilakukan untuk mengetahui apakah sistem berjalan dengan baik atau tidak. Pengujian bertujuan untuk mencari kesalahan. Pada tahap *testing*/pengujian aplikasi ini menggunakan Metode *Black Box*. *Black Box Testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program (Pressman,2007). Pada black box testing terdapat jenis teknik disain tes yang dapat dipilih berdasarkan pada tipe testing yang akan digunakan, yang diantaranya :

- 1) *functional Testing*
- 2) *Usability Testing*
- 3) *Recovery Testing*
- 4) *User acceptance Testing*
- 5) *Alpha Testing*
- 6) *Beta Testing*

Pada aplikasi ini menggunakan pengujian blackbox dengan tipe *functional testing*. Perangkat lunak diuji untuk persyaratan fungsional. Pengujian dilakukan dalam bentuk tertulis untuk memeriksa apakah aplikasi berjalan seperti yang diharapkan. Walaupun pengujian fungsional sering dilakukan dibagian akhir dari siklus pengembangan, masing – masing komponen dan proses dapat diuji pada awal pengembangan, bahkan sebelum sistem berfungsi, pengujian ini sudah dapat dilakukan pada seluruh sistem. Pengujian fungsional meliputi seberapa baik sistem melaksanakan fungsinya, termasuk perintah – perintah pengguna, manipulasi data, pencarian dan proses bisnis, pengguna layar, dan integrasi. Pengujian fungsional juga meliputi permukaan yang jelas dari jenis fungsi – fungsi, serta operasi *back-end* seperti, keamanan dan bagaimana meningkatkan sistem (Janner Simartama, 2010).

Dengan menggunakan tipe *Black Box* ini kebenaran sistem yang diuji dilihat berdasarkan *output* yang ada dari data atau kondisi masukkan yang diberikan untuk fungsi yang ada. Tanpa melihat proses, dari *output* tersebut dapat diukur dan diketahui kesalahan dari program dalam memenuhi kebutuhan *user*. Pengujian dilakukan menggunakan skenario proses yang telah ditentukan pada tabel 3.1 untuk user dan tabel 3.2 untuk admin.

Tabel 3.1 Kriteria Pengujian Kebutuhan Fungsional User

| No | Fungsi | Skenario Proses | Sistem Bekerja | Hasil Proses | Keterangan |
|----|--------|---------------------------------------------------|----------------|--------------|------------|
| 1 | Menu | User membuka halaman beranda, daftar, masuk/login | | | |

| | | | | | |
|---|-------------|----------------------------------------------------------------------------------|--|--|--|
| 2 | Create akun | User mengisi username, email, dan password pada form create akun yang disediakan | | | |
| 3 | Create akun | User mengisi username saja pada form create akun yang sudah disediakan | | | |
| 4 | Create akun | User mengisi username dan email saja | | | |
| 5 | Login | User login menggunakan username dan password yang benar | | | |
| 6 | Login | User melakukan login dengan username yang benar dan password yang salah | | | |
| 7 | Login | User melakukan login dengan username yang salah dan password yang benar | | | |
| 8 | Login | User melakukan login dengan username yang salah dan password yang salah | | | |

| | | | | | |
|----|------------------------------|------------------------------------------------------------------------------------|--|--|--|
| 9 | Lupa password | User lupa password dengan mengklik lupa password | | | |
| 10 | Form pendaftaran | User mengisi dengan benar semua form pendaftaran yang disediakan lalu tekan simpan | | | |
| 11 | Form pendaftaran | User tidak mengisi bagian penting yang ada di form pendaftaran lalu tekan simpan | | | |
| 12 | Perbaharui data | user memperbaharui data diri yang sudah tersimpan | | | |
| 13 | Mencetak formulir pembayaran | User mencetak formulir pembayaran | | | |
| 14 | Mencetak biodata | User mencetak biodata yang sudah tersimpan | | | |
| 15 | Logout | User menekan menu logout | | | |

Tabel 3.2 Kriteria Pengujian Kebutuhan Fungsional Admin

| No | Fungsi | Skenario Proses | Sistem Bekerja | Hasil Proses | Keterangan |
|----|--------|-------------------------|----------------|--------------|------------|
| 1 | Login | admin login menggunakan | | | |

| | | | | | |
|---|-------------------------|-------------------------------------------------------------------------------------|--|--|--|
| | | username dan password yang benar | | | |
| 2 | Login | admin melakukan login dengan username yang benar dan password yang salah | | | |
| 3 | Login | admin melakukan login dengan username yang salah dan password yang benar | | | |
| 4 | Login | Admin melakukan login dengan username yang salah dan password yang salah | | | |
| 5 | Kelola Admin | Admin menambahkan akun baru dan menghapus akun | | | |
| 6 | Pendaftan | Admin melihat pendaftar yang sudah masuk | | | |
| 7 | Kelola data calon siswa | Admin mengupdate data calon siswa jika diperlukan dengan mengklik update | | | |
| 8 | Kelola data calon siswa | Admin mendelete data calon siswa dengan mengklik delete | | | |
| 9 | Kelola data calon siswa | Admin melakukan registrasi pendaftar menjadi calon siswa dengan mengklik registrasi | | | |

| | | | | | |
|----|-------------------------------|-----------------------------------------------------------------------------|--|--|--|
| 10 | Kelola data calon siswa | Admin dapat merubah kembali status calon siswa menjadi pendaftar | | | |
| 11 | Kelola data calon siswa | Admin dapat melihat daftar calon siswa yang sudah valid di menu calon siswa | | | |
| 12 | Kelola data calon siswa | Admin dapat mengexport data calon siswa yang sudah valid kefile excel | | | |
| 13 | Kelola kelas | Admin dapat melakukan pembagian kelas | | | |
| 14 | Kelola data calon siswa | Admin dapat menghapus data calon siswa di menu calon siswa | | | |
| 15 | Kelola data calon siswa | Admin dapat mengupdate data calon siswa di menu calon siswa | | | |
| 16 | Kelola data calon siswa | admin dapat membuat pengumuman tentang PSB | | | |
| 17 | Kelola pengumuman atau berita | Admin dapat mengupdate pengumuman | | | |
| 18 | Kelola pengumuman atau berita | Admin dapat menghapus pengumuman | | | |
| 19 | Kelola fitur pendaftaran | Admin dapat mengatur fitur pendaftaran | | | |

| | | | | | |
|----|----------------|--------------------------------------------------------------------------------------|--|--|--|
| 20 | Kelola kuota | Admin dapat menentukan kouta calon siswa di setiap jenjang | | | |
| 21 | Kelola kuota | Admin dapat mengubah kouta disetiap jenjang | | | |
| 22 | Kelola seleksi | Admin dapat menentukan batas minimum nilai pada proses seleksi jenjang SMP dan SMK | | | |
| 23 | Kelola seleksi | Admin dapat merubah batas nilai minimum pada proses seleksi pada jenjang SMP dan SMK | | | |
| 24 | Kelola seleksi | Admin dapat menentukan batas umur minimum pada seleksi jenjang TK dan SD | | | |
| 25 | Kelola seleksi | Admin dapat merubah batas umur minimum pada seleksi jenjang TK dan SMP | | | |
| 26 | Lihat grafik | Admin dapat melihat grafik pendatar yang sudah masuk | | | |
| 27 | Logout | Admin mengklik Logout | | | |

Tabel skenario pengujian adalah tabel yang berisi instrumen yang akan diuji, instrumen yang di uji pada format ini adalah pengujian dari segi fungsi

perangkat lunak yang dibuat. Dari hasil skenario pengujian yang dijalankan akan menghasilkan hasil yang diharapkan si pengguna perangkat lunak, jika sistem bekerja dan sesuai maka perangkat lunak dikatakan berhasil dibuat tanpa adanya *bug* atau *error*.

3.5.5 Pemeliharaan

Pemeliharaan dianggap perlu dilakukan untuk memperbaiki jika ada kesalahan sistem yang tidak berjalan sesuai dengan fungsinya yang mungkin disebabkan karena perubahan tampilan atau perubahan kode program. Walaupun sudah diuji sebelumnya.

BAB IV

HASIL PENELITIAN

4.1. Hasil Analisis Dan Pembahasan

Pendaftaran, penyeleksian dan pengelompokan kelas siswa baru di YPI As Sa'adah Pondok Kelapa menjadi keunggulan aplikasi penerimaan siswa baru, sehingga memudahkan admin atau panitia dalam hal mengelola data calon siswa. Pada tahap seleksi ada dua aturan yaitu dengan standar umur untuk jenjang TK dan SD, dan nilai SKHUN untuk jenjang SMP dan SMK. Penerimaan Siswa Baru (PSB) merupakan kegiatan yang amat penting karena proses tersebut mempunyai nilai strategis guna menjaring calon siswa yang berkualitas. Dengan alasan tersebut, menjadikan aplikasi penerimaan siswa baru layak untuk dikembangkan. Berikut adalah beberapa analisis masalah pada BAB III yang harus dijawab oleh *software*:

- 1) Cara mengotomasi proses akses penerimaan siswa baru.
- 2) Memberikan akses pada admin untuk memasukan ,merubah, dan menghapus data pendaftar. Dan memberikan akses kepada siswa baru untuk mendaftar sesuai dengan jenjang.
- 3) Cara melihat alur pendaftar
- 4) Cara menginput data pribadi calon siswa.
- 5) Cara melihat pengumuman seleksi
- 6) Cara menampilkan resi pembayaran.
- 7) Cara mencetak bukti pendaftaran.

Berikut adalah hasil dari pengujian aplikasi atas permasalahan tersebut:

- 1) Masalah mengotomasi proses akses aplikasi penerimaan siswa baru dijawab dengan adanya sistem *login* dan hak akses pada proses masuk. Siswa mengakses aplikasi sebagai *user* dan pihak yayasan mengakses sebagai *admin*.
- 2) Pemberian hak akses pada admin untuk memasukan, merubah, dan menghapus setiap data yang dibutuhkan calon siswa
- 3) Cara kerja penginputan data pribadi pendaftar dan data yang dibutuhkan calon siswa pada *software* adalah dengan cara memilih menu pendaftaran dan menu biodata untuk merubah yang salah pada *form* yang sudah disediakan. Data yang sudah terinput pada *form* akan tersimpan pada *database*.
- 4) Untuk bisa mendaftar calon siswa harus melakukan create akun. Setelah mempunyai akun pendaftar melakukan pendaftaran dengan mengisi biodata terlebih dahulu. Jika ada kesalahan maka pendaftar harus login terlebih dahulu, setelah itu pendaftar bisa merubah kesalahan pada data pribadi.
- 5) Setelah mengisi biodata calon siswa harus mencetak bukti pendaftaran atau formulir sebagai bukti pendaftaran dan untuk keperluan validasi berkas.
- 6) Calon siswa mencetak resi pembayaran setelah mengisi formulir yang berisi data pribadi. Cetak
- 7) Calon siswa melihat pengumuman dengan melakukan login terlebih dahulu dan calon siswa harus melakukan validasi berkas

dan membayar pendaftaran. Jika belum melakukan validasi berkas maka tidak ada dalam pengumuman.

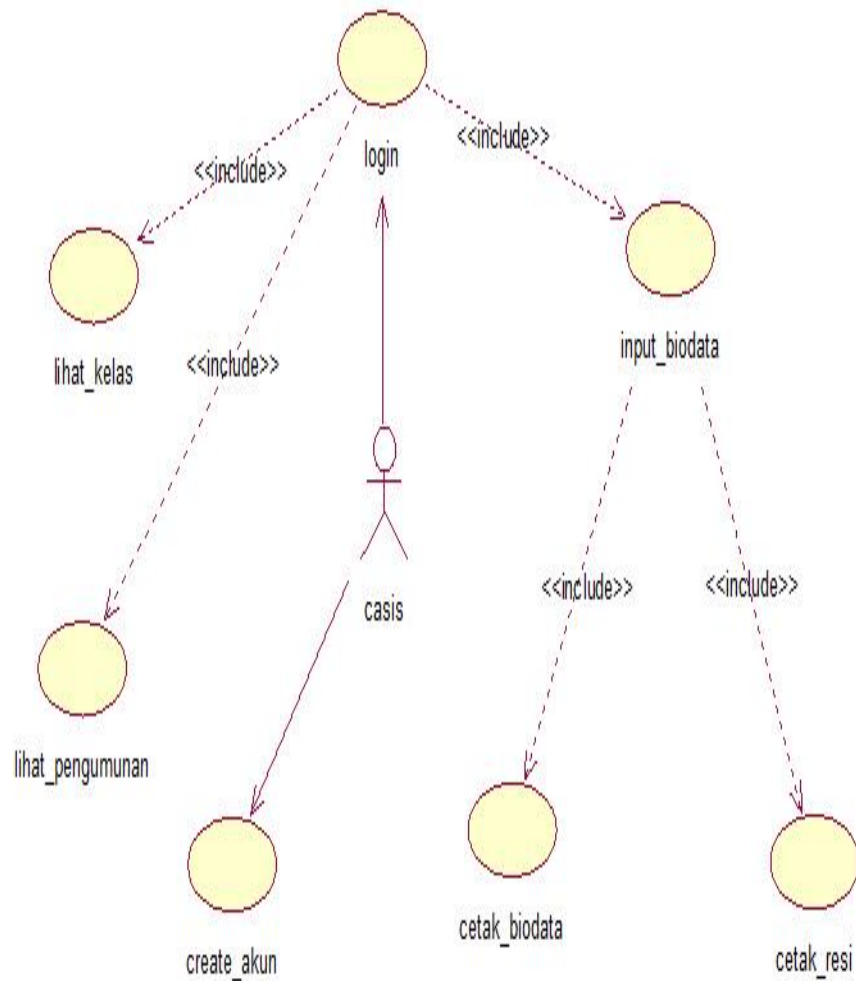
4.2. Perancangan

Tahap desain pada penelitian ini adalah membuat desain rancangan halaman – halaman yang diperlukan dalam aplikasi penerimaan siswa baru ini. Seperti yang telah disampaikan pada bab III. Setelah itu membuat rancangan dokumentasi sistem dengan menggunakan *Unified Modeling Language* (UML) yaitu dengan dibuatnya usecase, sequen diagram dan class diagram.

4.2.1 Usecase

Usecase Diagram secara grafis menggambarkan interaksi antara sistem, sistem eksternal, dan pengguna. Dengan kata lain *Usecase* diagram secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna (*user*) mengharapkan interaksi dengan sistem itu. *Usecase* secara naratif digunakan untuk secara tekstual menggambarkan sekuensi langkah-langkah dari setiap interaksi. Berikut ini adalah *Usecase* dari sistem PSB YPI As Sa'adah Pondok Kelapa :

2. Usecase Calon Siswa



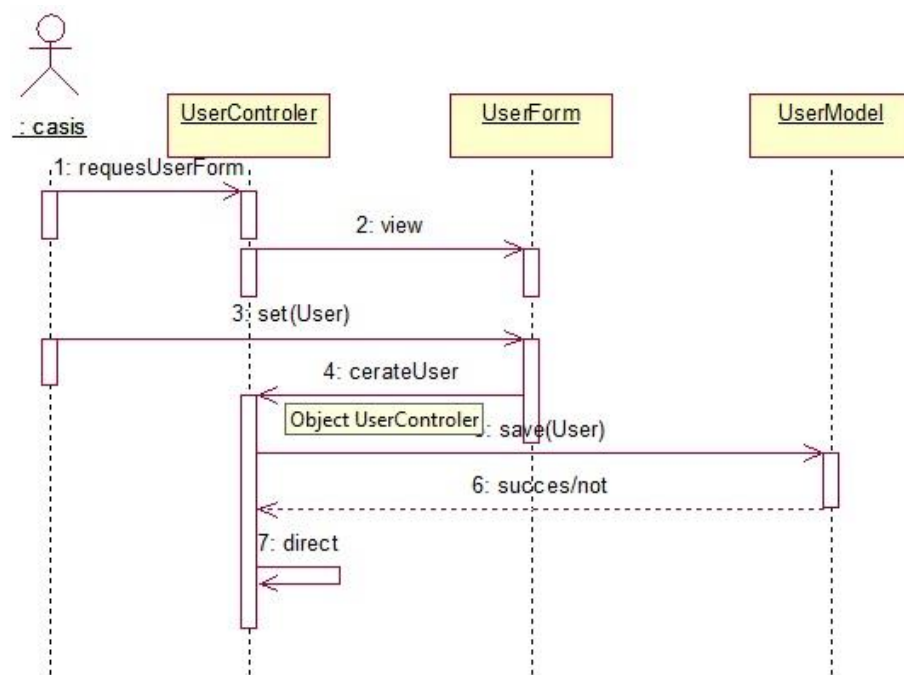
Gambar 4.2 Usecase Calon Siswa

Pada gambar diatas calon siswa sebagai *actor*. Siswa terlebih dahulu harus membuat akun dan calon siswa bisa melakukan *login*. Setelah itu calon siswa mengisi formulir dengan benar *printout* resi pembayaran dan bukti pendaftaran untuk pembayaran dan bukti pendaftaran.

4.2.2 Sequence Diagram

Sequence diagram menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah *usecase* atau operasi. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara objek dan dalam sekuensi. Berikut ini adalah *sequence diagram* dari sistem penerimaan siswa baru YPI As Sa'adah Pondok Kelapa :

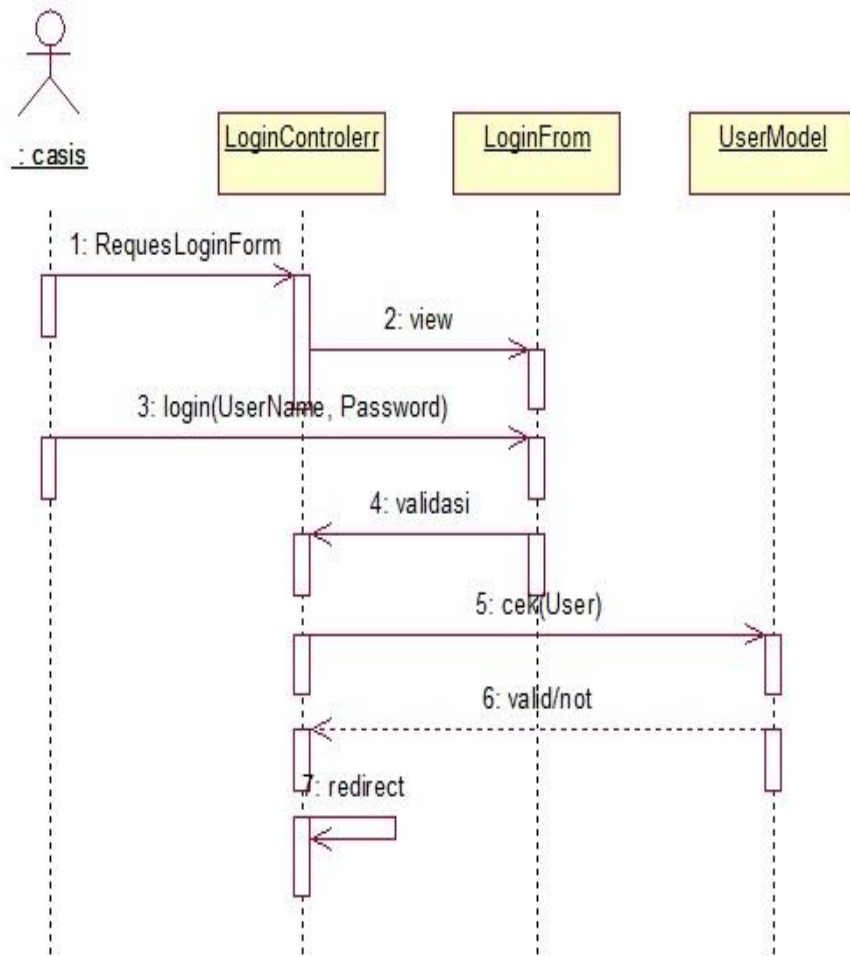
1. Sequence Diagram Create Akun Calon Siswa



Gambar 4.3 Sequence Diagram Create Akun Calon Siswa

Dari gambar diatas proses *sequence diagram* dimulai dari calon siswa meminta *form user* dari *user controller* lalu *user controller* mengirim pesan kepada *user form* untuk menampilkan. Calon siswa mengisi biodata *user* dan *mensubmit* biodata lalu *user model* menyimpan biodata kedalam *database*. Setelah itu *User controller* mengarahkan ke success page atau error.

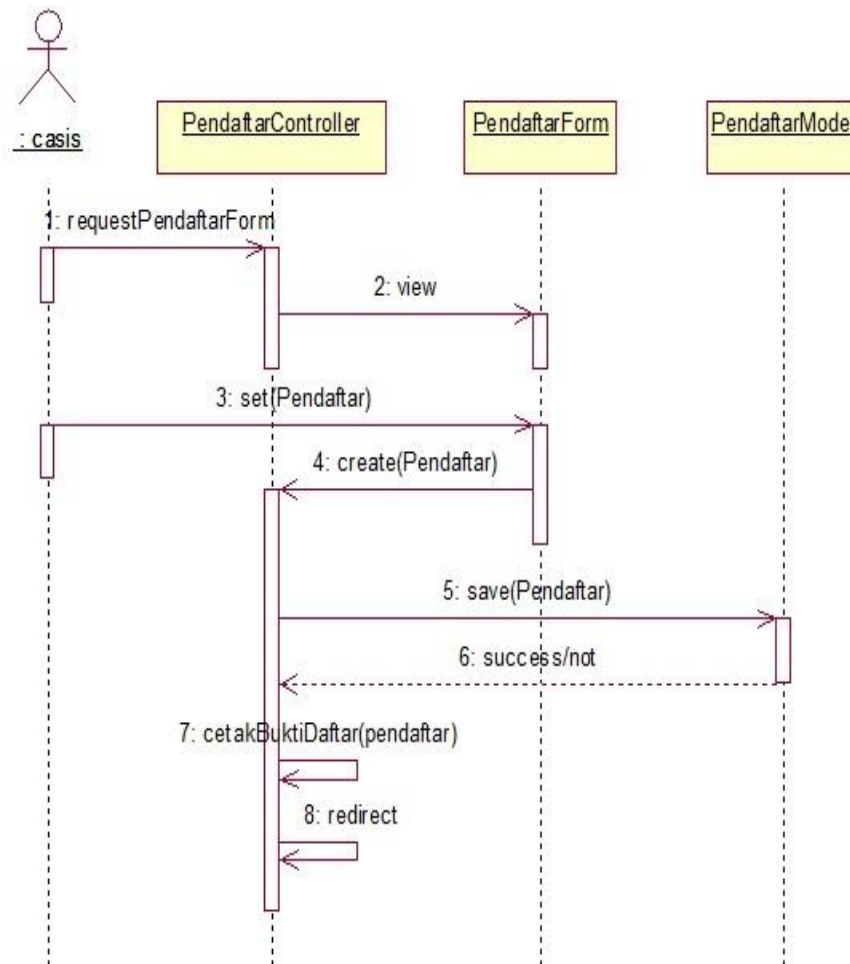
2. Sequence Diagram Login Calon Siswa



Gambar 4.4 Sequence Diagram Login Siswa

proses *sequence diagram* dimulai dari calon siswa meminta *form login* dari *login controller*. Lalu *login controller* mengirim pesan kepada *login form* untuk menampilkan *form login*. Calon siswa mengisi *form login* dengan *username* dan *password* yang sudah dibuat sebelumnya dan mensubmit. *Login controller* mengecek kepada *user model* dan mengarahkan ke *success* atau *error*.

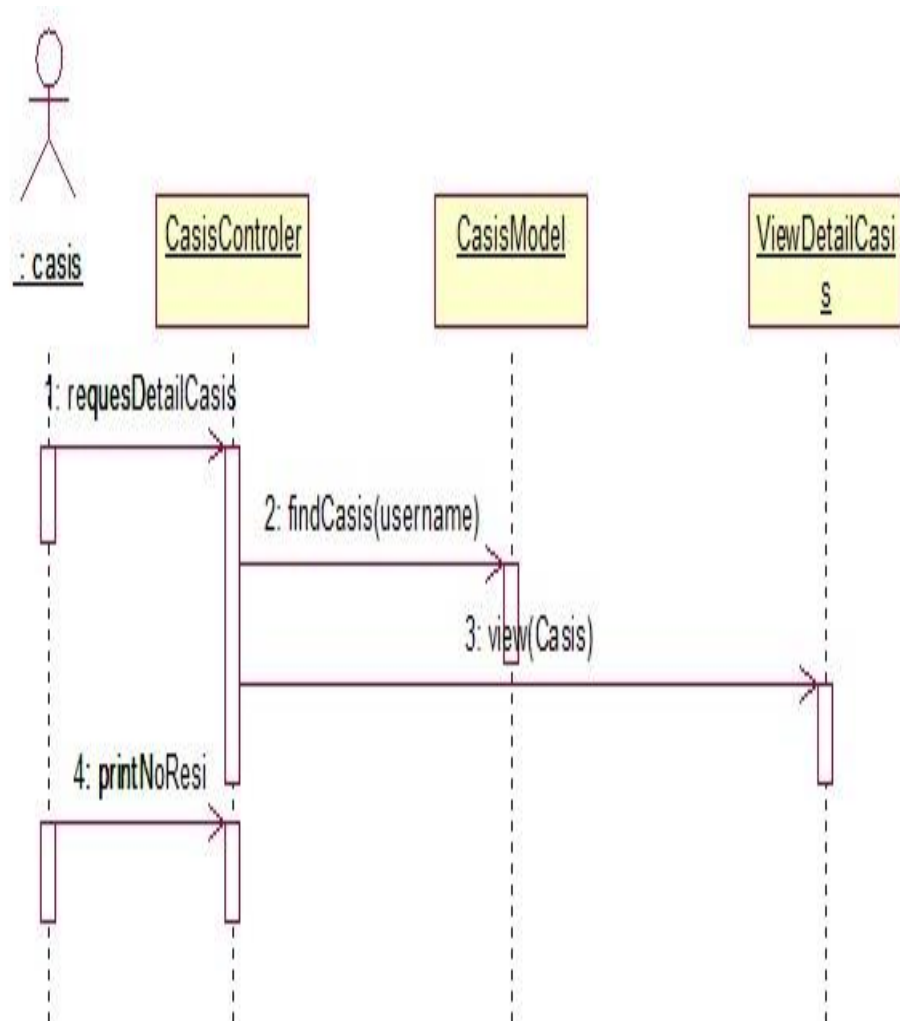
3. Sequence Diagram Daftar Diri Siswa



Gambar 4.5 Sequence Diagram Daftar Diri Siswa

pada gambar *sequence diagram* diatas calon siswa meminta *form* pendaftaran dari *pendaftar controller*. Lalu *pendaftar controller* mengirim pesan kepada *pendaftar form* untuk menampilkan. Calon siswa mengisi *form* pendaftaran dan *mensubmit*. *Pendaftar model* menyimpan biodata *pendaftar* dan *pendaftar controller* mengarahkan ke halaman *success* atau *error*. *Pendaftar controller* juga mengarahkan ke halaman cetak bukti pendaftaran.

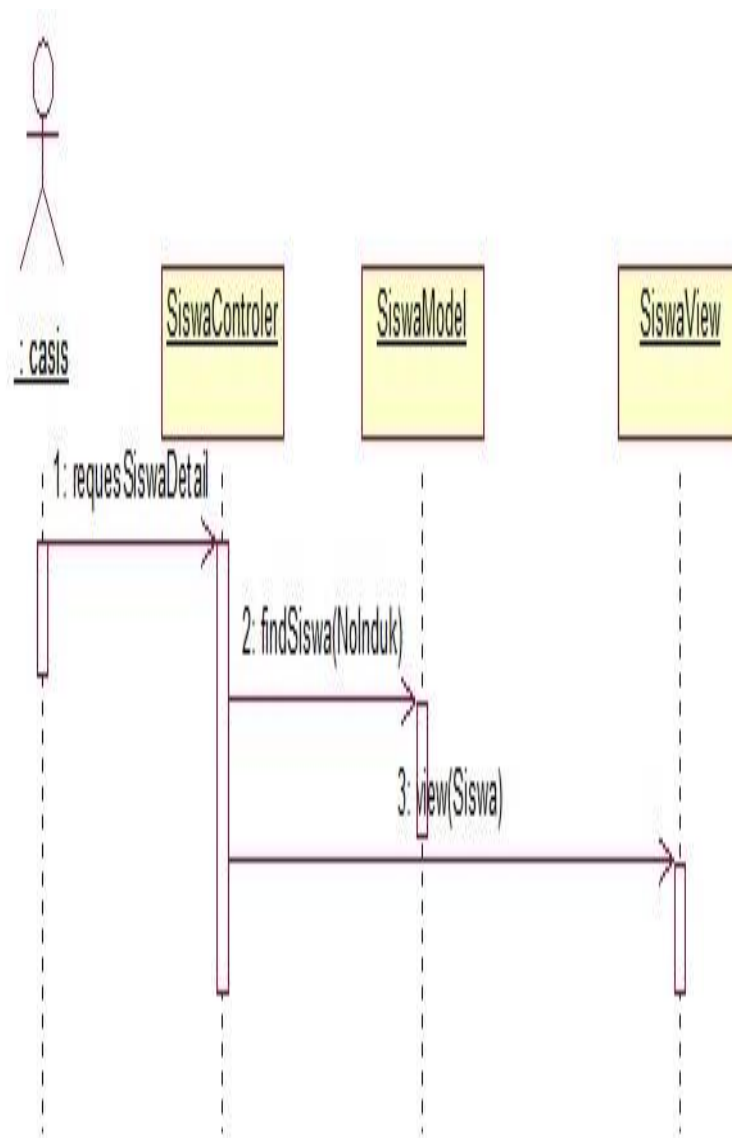
4. *Sequence Diagram Cetak Resi*



Gambar 4.6 *Sequence Diagram Cetak Resi*

pada gambar *sequence diagram* diatas proses dimulai dari calon siswa meminta detail casis dari casis *controller*. Lalu casis *controller* mengirim pesan ke casis model untuk mencari dengan username tersebut. Casis *controller* mengirim pesan ke *view detail* casis untuk menampilkan *form* cetak resi. Sebelum mencetak resi sistem mencari detail *user* dan menampilkan detail *user*. Setelah itu calon siswa *printout* resi pembayaran.

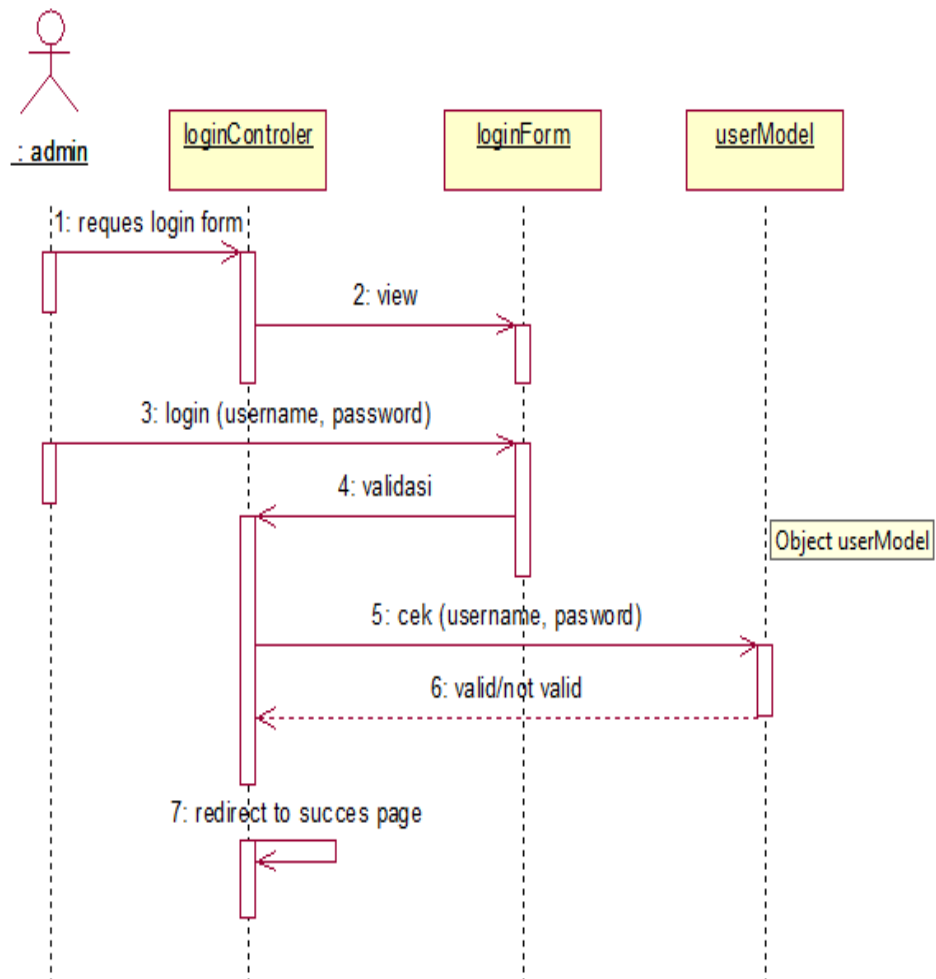
5. Sequence Diagram Lihat Kelas



Gambar 4.7 Sequence Diagram Lihat Kelas

pada gambar proses *sequence diagram* diatas casis meminta detail siswa dari siswa *controller*, lalu siswa *controller* mengirim pesan kepada siswa *model* untuk mencari detail siswa dengan nomor induk tersebut. Setelah selesai siswa *controller* mengirim pesan kepada siswa *view* untuk menampilkan.

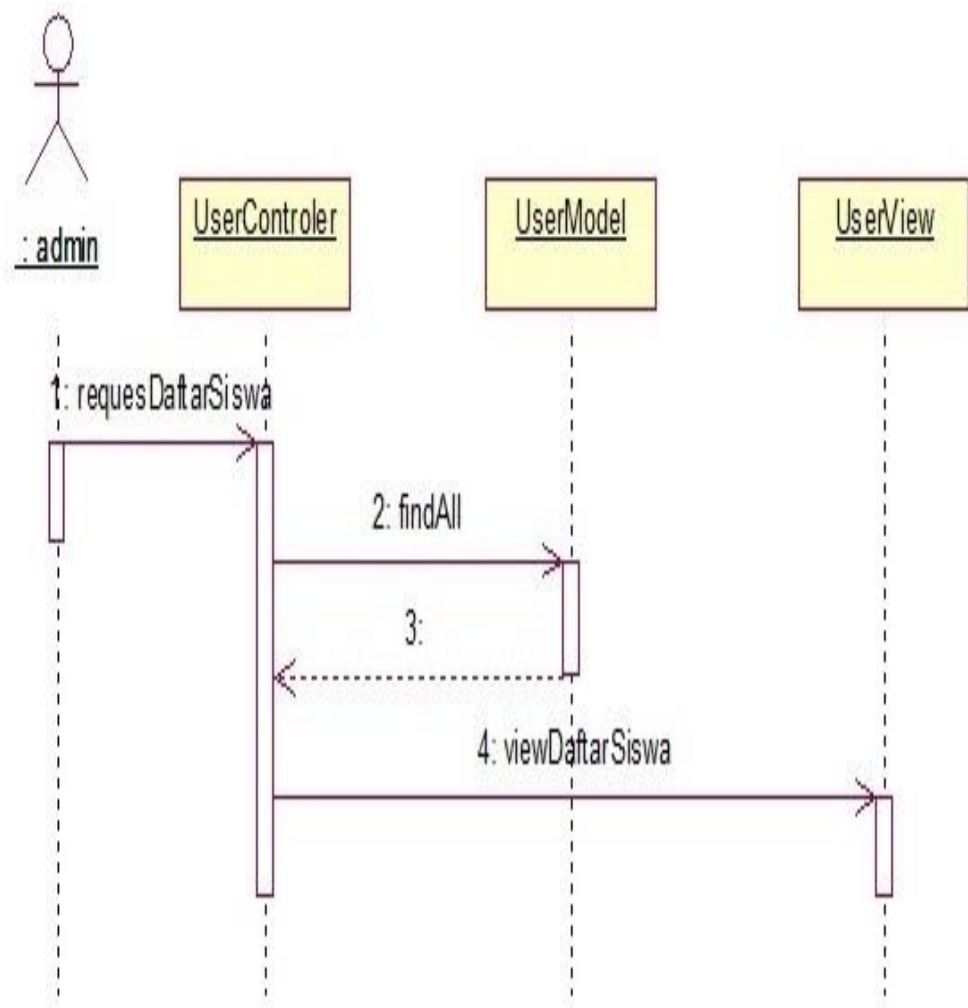
6. Sequence Diagram Login Admin



Gambar 4.8 Sequence Diagram Login Admin

Pada *sequence diagram* dimulai dari admin meminta *form login* dari login controller. Lalu *login controller* mengirim pesan kepada *login form* untuk menampilkan halaman *login*. Admin mengisi *form login* dengan *username*, *password* dan *submit*. Setelah itu *login controller* mengirim pesan kepada *user model* untuk mengecek *username*, *password* apakah valid atau tidak. *Login controller* mengarahkan ke halaman *success* atau *error*.

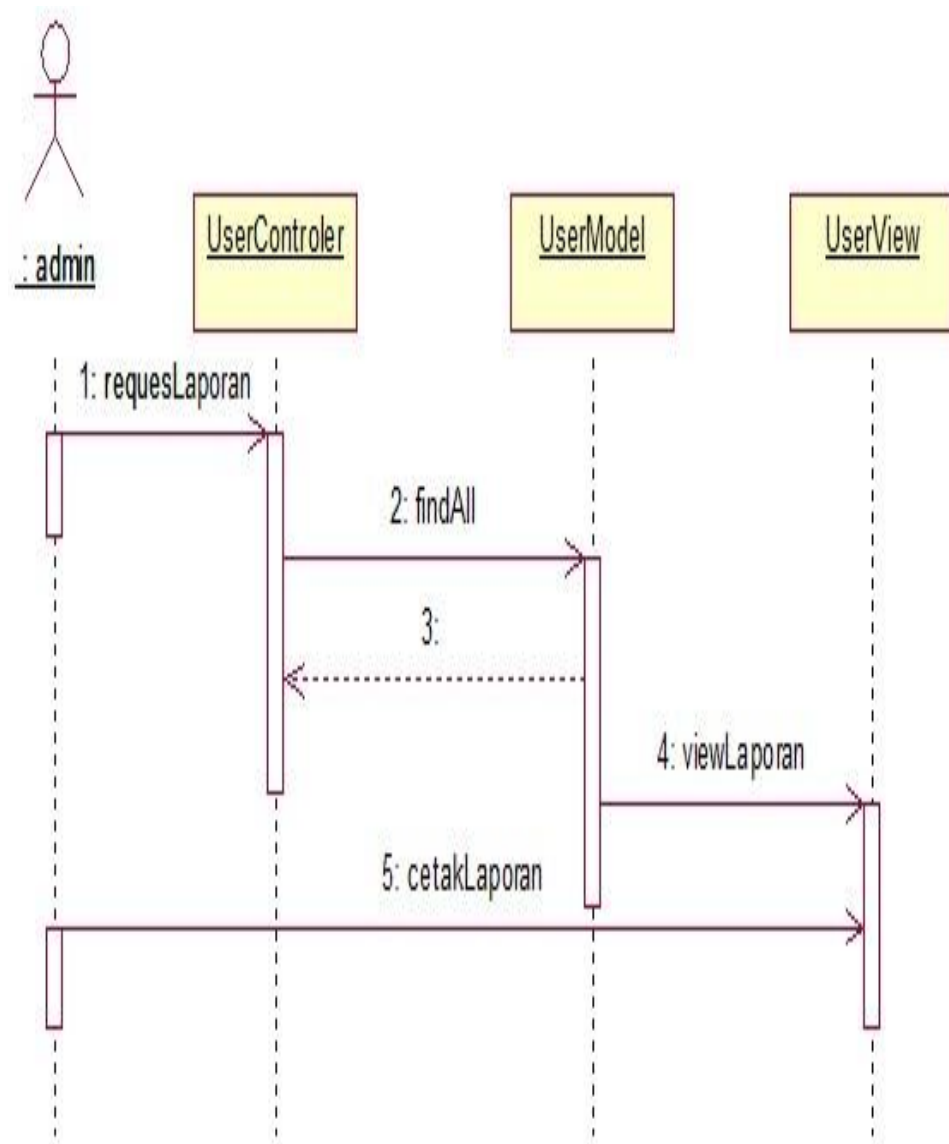
7. Sequence Diagram Daftar Siswa Admin



Gambar 4.9 Sequence Diagram Daftar Siswa Admin

Pada sequence diagram diatas Admin meminta data siswa dari *user controler*. Lalu *user controller* mengirim pesan kepada *user model* untuk mencari data siswa yang telah masuk di sistem. *User controller* mengirim pesan kepada *user view* untuk memampikan dseluruh data siwa yang sudah mendaftar. Admin bisa melihat, mengubah, dan menghapus sesuai dengan ketentuan yang berlaku, jika ada masalah dalam proses pendaftaran..

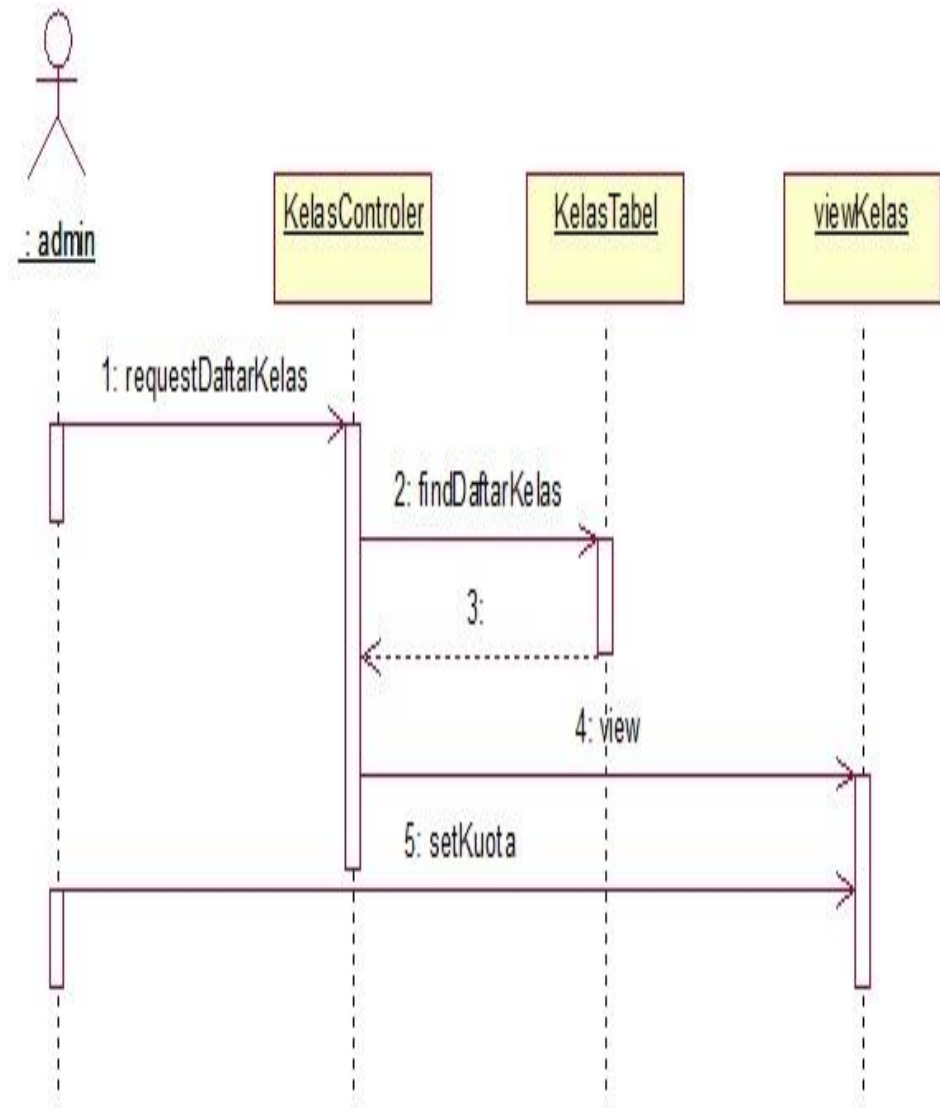
8. *Sequence Diagram Cetak/Lihat Laporan Admin*



Gambar 4.10 *Sequence Diagram Cetak Laporan*

Pada proses cetak laporan ini admin berhak untuk mengetahui semua pendaftar yang sudah masuk di sistem. Admin meminta laporan dari *user controller*. Lalu *user controller* mengirim pesan kepada *user tabel* untuk mencari laporan. Setelah itu *user tabel* mengirim pesan kepada *user view* untuk menampilkan laoporan. Admin melihat laporan dan mencetak laporan.

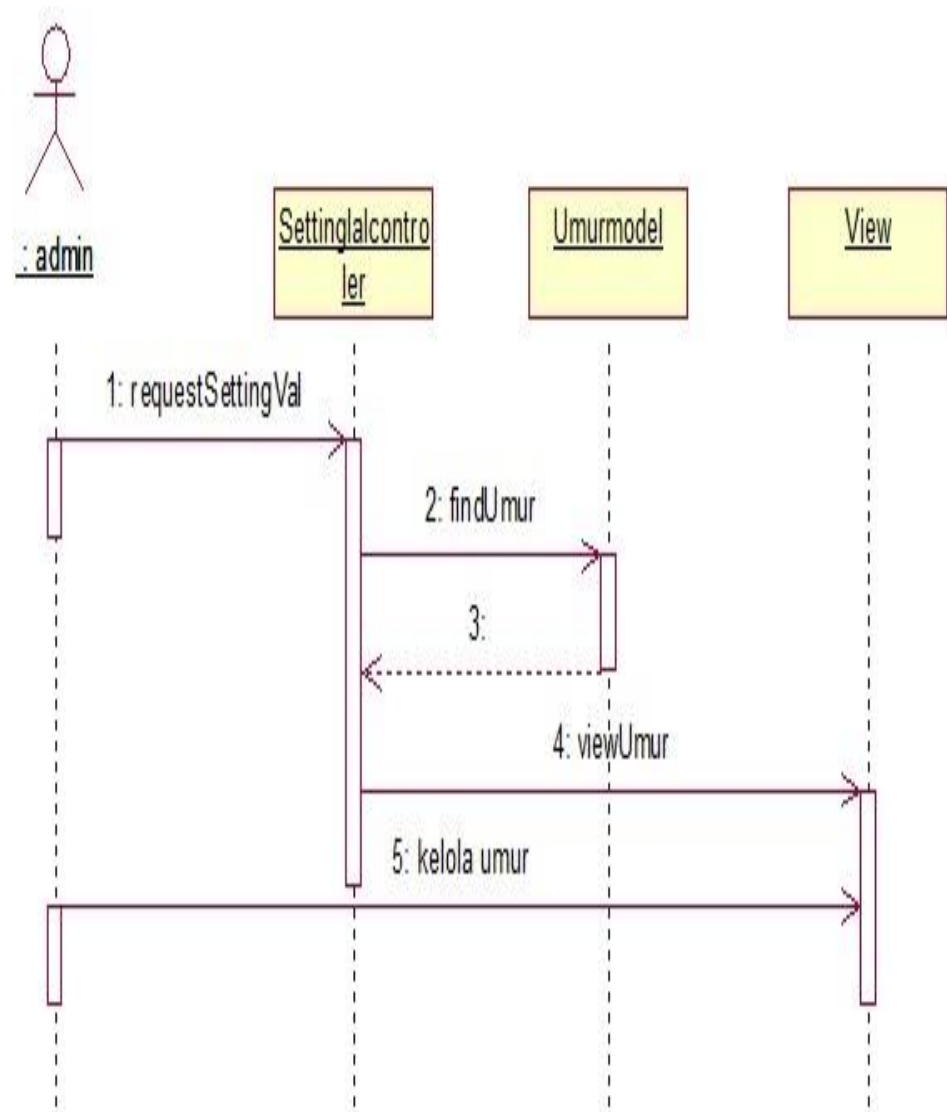
9. Sequence Diagram Bagi Kelas



Gambar 4.11 Sequence Diagram Bagi Kelas

Pada gambar sequence diagram bagi kelas diatas admin meminta daftar kelas dari kelas controller. Lalu kelas controller mengirim pesan kepada kelas tabel untuk mencari. Selanjutnya kelas controller mengirim pesan kepada view untuk menampilkan. Admin mengelola kuota sesuai yang dibutuhkan oleh sekolah.

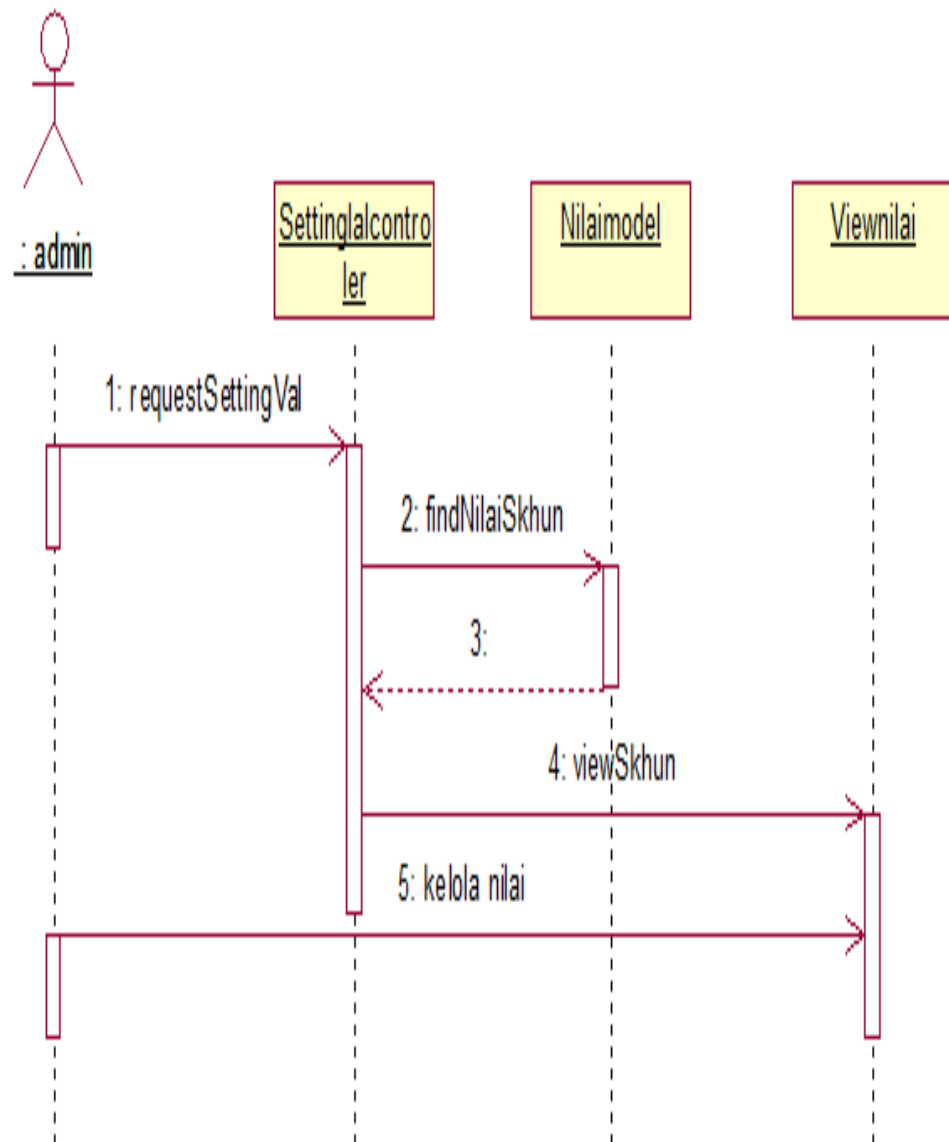
10. Sequen Diagram Kelola Umur Seleksi Admin



Gambar 4.12 Sequence Diagram Setting Value Umur

pada proses *setting value* umur ini berfungsi untuk melakukan seleksi di jenjang TK dan SD. Admin meminta *setting value* dari *setting controller*. Lalu *setting controller* mengirim pesan kepada umur *model* untuk mencari. Setelah itu *setting controller* mengirim pesan kepada *view* untuk menampilkan halaman *setting value* dan admin mengelola umur untuk menentukan seleksi.

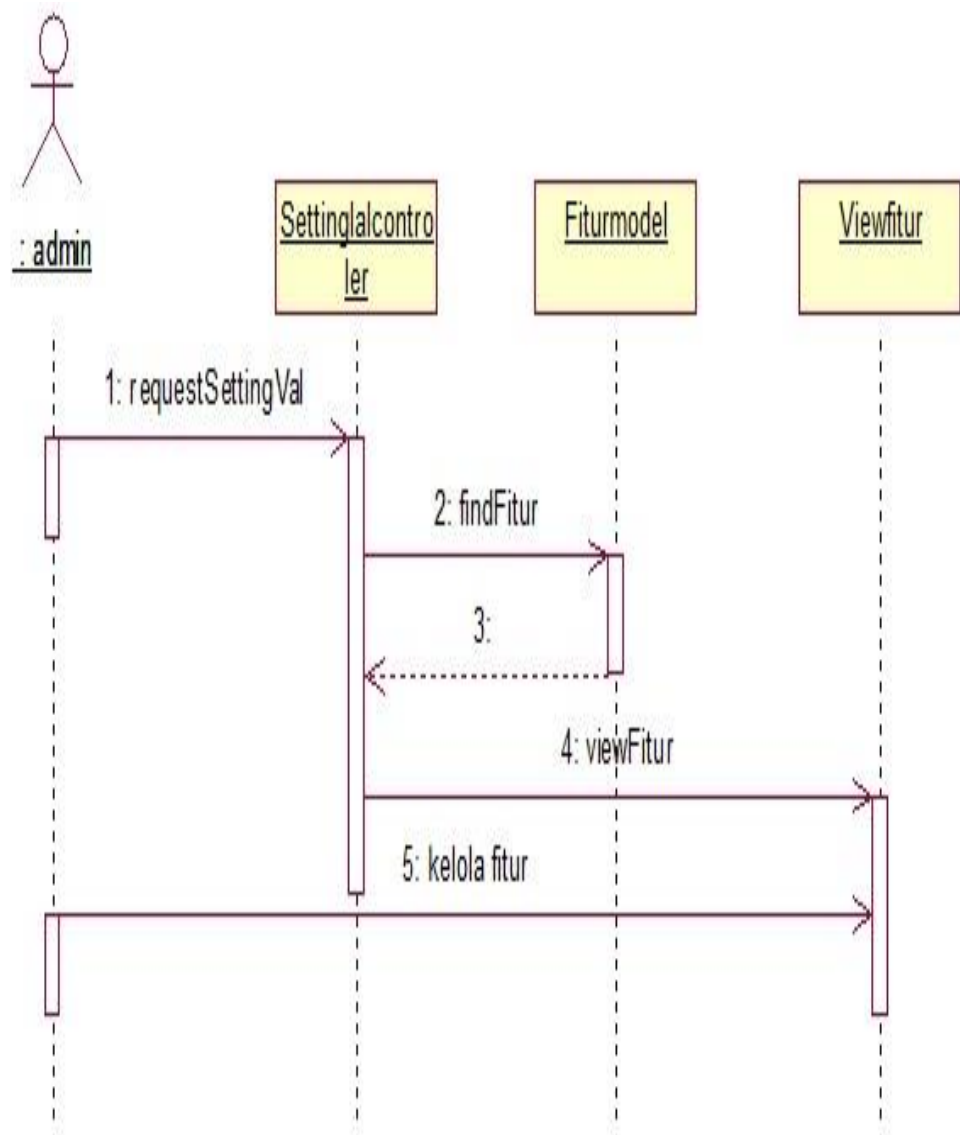
11. Sequence Diagram Kelola Nilai Seleksi Admin



Gambar 4.13 Sequence Diagram Setting Value Nilai

pada proses *setting value* nilai ini berfungsi untuk melakukan seleksi di jenjang SMP dan SMK. Admin meminta *setting val* pada *setting controller*. lalu *setting controller* mengirim pesan kepada *nilai model* untuk mencari. *Setting controller* mengirim pesan kepada *view nilai* untuk menampilkan. setelah itu admin mengelola nilai sesuai dengan kebutuhan untuk proses seleksi.

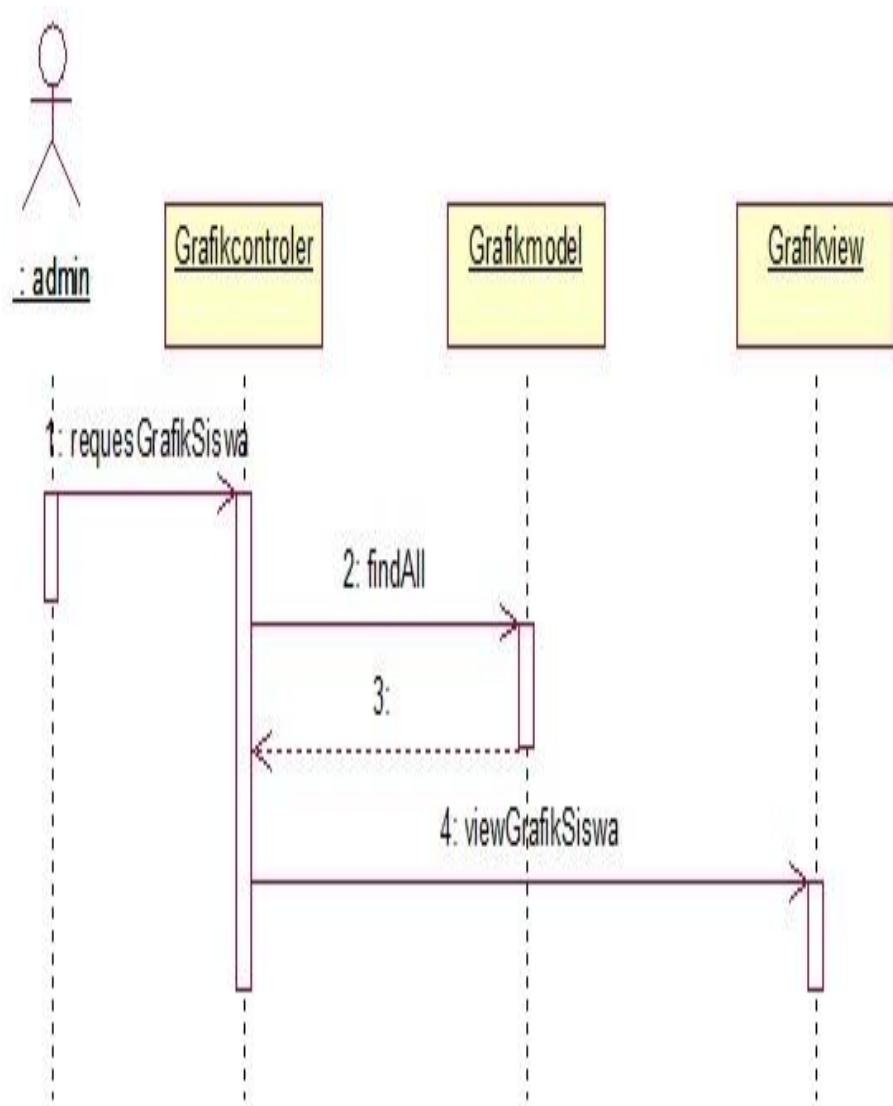
12. Sequence Diagram Atur Fitur Admin



Gambar 4.14 Sequence Diagram Atur Fitur

pada proses *setting value* fitur ini berfungsi untuk mengatur proses pendaftaran dibuka dan ditutup. Admin meminta atur fitur dari setting controller. Lalu setting controller mengirim pesan kepada fitur model untuk mencari. Setelah itu setting controller mengirim pesan kepada view untuk menampilkan halaman kelola fitur dan admin mengelola fitur sesuai dengan kebutuhan.

13. Sequence Diagram Lihat Grafik Admin



Gambar 4.15 Sequence Diagram Lihat Grafik Admin

pada proses lihat grafik berfungsi untuk mengetahui jumlah pendafar yang sudah masuk dari jenjang TK, SD, SMP, dan SMK. Admin meminta grafik siswa dari grafik controller. Lalu grafik controller mengirim pesan kepada crafik model untuk mencari. Grafik controller mengirim pesan kepada view untuk menampilkan grafik calon siswa yang sudah mendaftarkan diri.

4.2.3.1 Daftar Class

Berikut ini adalah daftar class diagram dari aplikasi penerimaan siswa baru:

1. User Class

Tabel 4.1 User Class

| <i>Atribute</i> | | |
|-----------------|-----------------------------------|----------|
| No | Nama | Type |
| 1 | <u>Id</u> | Int |
| 2 | Username | Varchar |
| 3 | Auth_key | Varchar |
| 4 | Password_hash | Varchar |
| 5 | Password_reset_token | Varchar |
| 6. | Email | Varchar |
| 7 | Status | Smallint |
| 8 | Created_at | Date |
| 9 | Upadate_at | Date |
| 10 | Unit | Tinyint |
| <i>Methode</i> | | |
| 1 | findByUsername(\$username) | |
| 2 | findByPasswordResetToken(\$token) | |
| 3 | setPassword(\$password) | |
| 4 | removePasswordResetToken() | |

2. Calon Siswa Class

Tabel 4.2 Casis Class

| <i>Attribute</i> | | |
|------------------|--------------------------------|---------|
| No | Nama | Type |
| 1 | <u>Id calon peserta</u> | Int |
| 2 | User_id | Int |
| 3 | No_peserta | Char |
| 4 | Photo | Char |
| 5 | Nama | Varchar |
| 6 | Nik | Char |
| 7 | Jns_kelamin | Char |
| 8 | Tmp_lahir | Varchar |
| 9 | Tanggal_lahir | Date |
| 10 | Alamat | Varchar |
| 11 | Agama | Char |
| 12 | Anak_ke | Varchar |
| 13 | Jumlah_saudara | Varchar |
| 14 | Yatim_piatu | Varchar |
| 15 | No_telp | Varchar |
| 16 | Kewarganegaraan | Varchar |
| 17 | Gol_darah | Char |
| 18 | Tinggibadan | Int |
| 19 | Beratbadan | Int |
| 20 | Penyakit | Varchar |
| 21 | Nisn | Varchar |
| 22 | Skhun | Varchar |
| 23 | No_sttb | Char |
| 24 | Sekolah_asal | Varchar |

| | | |
|-----------------------|--------------------|---------|
| 25 | Nama_ayah | Varchar |
| 26 | Tempat_lahir_ayah | Varchar |
| 27 | Tanggal_lahir_ayah | Date |
| 28 | Pekerjaan_ayah | Char |
| 29 | Pendidikan_ayah | Varchar |
| 30 | Nama_ibu | Varchar |
| 31 | Tempat_lahir_ibu | Varchar |
| 32 | Tanggal_lahir_ibu | Date |
| 32 | Pendidikan_ibu | Varchar |
| 34 | Pekerjaan_ibu | Char |
| 35 | Alamat_ortu | Varchar |
| 36 | Penghasilan_ortu | Char |
| 37 | Nama_wali | Varchar |
| 38 | Tempat_lahir_wali | Varchar |
| 39 | Tanggal_lahir_wali | Date |
| 40 | Pendidikan_wali | Varchar |
| 41 | Pekerjaan_wali | Char |
| 42 | Alamat_wali | Varchar |
| 43 | Penghasilan_wali | Char |
| <i>Methode</i> | | |
| 1 | getUser() | |
| 2 | getPassed() | |
| 3 | register() | |
| 4 | getAge() | |
| 5 | getUnit() | |
| 6 | getNis() | |
| 7 | actionIndex() | |
| 8 | actionView() | |

3. Pembayaran Psb Class

Tabel 4.3 Pembayaran Psb Class

| <i>Attribute</i> | | |
|------------------|---------------------------------|--------------|
| No | Nama | Type |
| 1 | <u>Id pembayaran psb</u> | Int (11) |
| 2 | Kasir | Int (11) |
| 3 | User_id | Int (11) |
| 4 | Tanggal_bayar | Datetime |
| 5 | Jumlah_bayar | Varchar (45) |
| 6 | Jenis_pembayaran | Char (1) |
| 7 | Nomorkwitansi | Char (12) |
| <i>Methode</i> | | |
| 1 | getUser() | |
| 2 | getKasir() | |

4. Calon Peserta Search Model Class

Tabel 4.4 Calon Peserta Search Model Class

| <i>Attribute</i> | | |
|------------------|--------------------------------|-----------|
| No | Nama field | Tipe Data |
| 1 | <u>Id calon peserta</u> | Int |
| 2 | User_id | Int |
| 3 | Nama | Varchar |
| 4 | Jns_kelamin | Char |
| 5 | Agama | Char |
| 6 | Gol_darah | Char |
| 7 | Alamat | Varchar |
| <i>Method</i> | | |
| 1 | Search | |

5. Angkatan Class

Tabel 4.5 Angkatan Class

| <i>Attribute</i> | | |
|------------------|---------------------------|----------|
| No | Nama | Type |
| 1 | <u>Id angkatan</u> | Int (3) |
| 2 | Angkatan | char (4) |
| <i>Method</i> | | |
| 1 | GetTblSiswas | |
| 2 | FindByTahun | |

6. Unit Class

Tabel 4.6 unit Class

| <i>Atrribut</i> | | |
|-----------------|-----------------------|---------|
| No | Nama | Type |
| 1 | <u>Id unit</u> | Int |
| 2 | Unit | Varchar |
| 3 | Kepala sekolah | Int |
| <i>Method</i> | | |
| 1 | GetUsers | |
| 2 | GetUnitList | |

7. Admin Class

Tabel 4.7 Admin Class

| <i>Attribute</i> | | |
|------------------|------------------------|---------|
| No | Nama | Type |
| 1 | <u>Id_admin</u> | int |
| 2 | Username | varchar |
| 3 | Password_hash | Varchar |
| 4 | Created_at | Int |
| 5 | Update_at | Int |
| 6 | Role | Char |
| <i>Methode</i> | | |
| 1 | GetId | |
| 2 | findByUsername | |
| 3 | SetPassword | |
| 4 | GetAuthKey | |
| 5 | validateAuthKey | |

8. Admin Search Model Class

Tabel 4.8 Admin Search Model Class

| <i>Attribute</i> | | |
|------------------|------------------------|---------|
| No | Nama | type |
| 1 | <u>Id_admin</u> | int |
| 2 | Username | varchar |
| 3 | Password_hash | Varchar |
| 4 | Created_at | Int |
| 5 | Update_at | Int |
| <i>Methode</i> | | |
| 1 | Search | |

9. Username Class

Tabel 4.9 Username Class

| <i>Attribute</i> | | |
|------------------|---------------------------|---------|
| No | Nama | Type |
| 1 | <u>Id_username</u> | Int |
| 2 | Username | Varchar |
| 3 | Password | char |
| 4 | username_rule | char |
| <i>Method</i> | | |
| 1 | GetPembayaranPsbs | |
| 2 | GetTblKelas | |
| 3 | GetTblPegawais | |
| 4 | GetTblUnits | |

10. Pegawai Class

Tabel 4.10 Pegawai Class

| <i>Attribute</i> | | |
|------------------|-------------------|---------|
| No | Nama | Type |
| 1 | <u>Nip</u> | Int |
| 2 | Nuptk | varchar |
| 3 | Nik | char |
| 4 | Nama_pegawai | Varchar |
| 5 | Tempat_lahir | Varchar |
| 6 | Tanggal_lahir | Date |
| 7 | Jenis_kel | Char |
| 8 | Alamat | Varchar |

| | | |
|----------------|---------------|------|
| 9 | Id_username | Int |
| 10 | Id_jabatan | Char |
| 11 | Photo | Char |
| <i>methode</i> | | |
| 1 | GetIdJabatan | |
| 2 | GetIdUsername | |

11. Siswa Class

Tabel 4.11 Siswa Class

| <i>atribute</i> | | |
|-----------------|--------------------------------|-----------|
| No | Nama field | Tipe Data |
| 1 | <u>Id_calon_peserta</u> | Int |
| 2 | User_id | Int |
| 3 | Nama | Varchar |
| 4 | Jns_kelamin | Char |
| 5 | Agama | Char |
| 6 | Gol_darah | Char |
| 7 | Kewarganegaraan | Varchar |
| 8 | Tinggibadan | Int |
| 9 | Beratbadan | Int |
| 10 | Penyakit | Varchar |
| <i>methode</i> | | |
| 1 | GetIdUsername | |
| 2 | GetKelas | |
| 3 | GetAngkatan | |

12. Pengumuman Search Model Class

Tabel 4.12 Pengumuman Search Model Class

| <i>Attribute</i> | | |
|------------------|-----------------------------|------------|
| No | Nama field | Tipe Data |
| 1 | <u>Id_pengumuman</u> | Int |
| 5 | Id_admin | Int |
| 3 | Photo | Char |
| 4 | Create_at | Datetime |
| 2 | Pengumuman | Longtext |
| 6 | Summary | Mediumtext |
| 7 | Title | Varchar |
| <i>Method</i> | | |
| 1 | Search | |

13. Pengumuman Class

Tabel 4.13 Pengumuman Class

| <i>Attribute</i> | | |
|------------------|-----------------------------|------------|
| No | Nama field | Tipe Data |
| 1 | <u>Id_pengumuman</u> | Int |
| 2 | Pengumuman | Longtext |
| 3 | Photo | Char |
| 4 | Create_at | Datetime |
| 5 | Id_admin | Int |
| 6 | Summary | Mediumtext |
| 7 | Title | Varchar |
| <i>Method</i> | | |
| 1 | GetIdAdmin | |

| | |
|---|--------------|
| 2 | UploadPhoto |
| 3 | GetImageFile |
| 4 | GetPhotoUrl |

14. Setting Fitur Class

Tabel 4.14 Setting Fitur Class

| <i>Attribute</i> | | |
|------------------|--------------------------|-----------|
| No | Nama field | Tipe Data |
| 1 | <u>Id_setting</u> | Int |
| 2 | Setting | Varchar |
| 3 | Status | Char |
| <i>Method</i> | | |
| 1 | Search | |

15. Set Pembayaran Class

Tabel 4.15 Set Pembayaran Class

| <i>Attribute</i> | | |
|------------------|---------------------------------|--------------|
| No | Nama field | Tipe Data |
| 1 | <u>Jumlah_pembayaran</u> | Varchar |
| 2 | Tanggal_pembayaran | datetime |
| 3 | Jenis_pembayaran | Char |
| 4 | Unit | Varchar (45) |
| <i>Method</i> | | |
| 1 | GetUnit | |

16. Login Form Class

Tabel 4.16 Login Form Class

| <i>Attribute</i> | | |
|------------------|------------------------|-----------|
| No | Nama field | Tipe Data |
| 1 | <u>Username</u> | varchar |
| 2 | Password | Varchar |
| 3 | Remembername | Varchar |
| 4 | User | Varchar |
| <i>Method</i> | | |
| | GetUser | |
| | Rule | |
| | ValidatePasswor | |
| | Login | |

17. Setting Value Class

Tabel 4.17 Setting Value Class

| <i>Attribute</i> | | |
|------------------|--------------------------|-----------|
| No | Nama field | Tipe Data |
| 1 | <u>Id setting</u> | Int |
| 2 | Setting | Varchar |
| 3 | Value | Varchar |
| <i>Method</i> | | |
| 1 | GetSDAgeReq | |
| 2 | GetTKAgeReq | |
| 3 | GetSMPSKHUNReq | |
| 4 | GetSMKSKHUNReq | |

18. Signup Form Class

Tabel 4.18 Signup Form Class

| <i>Attribute</i> | | |
|------------------|------------------------|-----------|
| No | Nama field | Tipe Data |
| 1 | <u>Username</u> | Varchar |
| 2 | Password | Char |
| 3 | Email | Varchar |
| 4 | Jenjang | Char |
| <i>Method</i> | | |
| 1 | Signup | |

4.2.4 Sruktur Database

1. Tabel Admin

Primary Ke : id_admin

Foreign Key : -

Tabel 4.19 Admin

| No | Nama field | Tipe data | Keterangan |
|----|------------------------|---------------|----------------|
| 1 | <u>Id admin</u> | int (11) | Auto_Increment |
| 2 | Username | varchar (45) | Not null |
| 3 | Password_hash | Varchar (255) | Not null |
| 4 | Created_at | Int (11) | Not null |
| 5 | Update_at | Int (11) | Not null |
| 6 | Role | Char (2) | Not null |

2. Tabel Calon Peserta

Primary Key : id_calon_peserta

Foreign Key : user_id

Tabel 4.20 Calon Peserta

| No | Nama field | Tipe Data | Keterangan |
|----|--------------------------------|--------------|----------------|
| 1 | <u>Id calon peserta</u> | Int (11) | Auto_Increment |
| 2 | User_id | Int (11) | Not null |
| 3 | No_peserta | Char (13) | Not null |
| 4 | Photo | Char (36) | Not null |
| 5 | Nama | Varchar (45) | Not null |
| 6 | Nik | Char (16) | Not null |
| 7 | Jns_kelamin | Char (1) | Not null |
| 8 | Tmp_lahir | Varchar (45) | Not null |
| 9 | Tanggal_lahir | Date | Not null |
| 10 | Alamat | Varchar (45) | Not null |
| 11 | Agama | Char (1) | Not null |
| 12 | Anak_ke | Varchar (2) | Not null |
| 13 | Jumlah_saudara | Varchar (2) | Not null |
| 14 | Yatim_piatu | Varchar (1) | Not null |
| 15 | No_telp | Varchar (20) | Not null |
| 16 | Kewarganegaraan | Varchar (2) | Not null |
| 17 | Gol_darah | Char (2) | Not null |
| 18 | Tinggibadan | Int (3) | Not null |
| 19 | Beratbadan | Int (3) | Not null |
| 20 | Penyakit | Varchar (45) | Not null |
| 21 | Nisn | Varchar (10) | Not null |
| 22 | Skhun | Varchar (50) | Not null |
| 23 | No_sttb | Char (10) | Not null |

| | | | |
|----|--------------------|---------------|----------|
| 24 | Sekolah_asal | Varchar (45) | Not null |
| 25 | Nama_ayah | Varchar (45) | Not null |
| 26 | Tempat_lahir_ayah | Varchar (150) | Not null |
| 27 | Tanggal_lahir_ayah | Date | Not null |
| 28 | Pekerjaan_ayah | Char (1) | Not null |
| 29 | Pendidikan_ayah | Varchar (2) | Not null |
| 30 | Nama_ibu | Varchar (45) | Not null |
| 31 | Tempat_lahir_ibu | Varchar (150) | Not null |
| 32 | Tanggal_lahir_ibu | Date | Not null |
| 32 | Pendidikan_ibu | Varchar (2) | Not null |
| 34 | Pekerjaan_ibu | Char (1) | Not null |
| 35 | Alamat_ortu | Varchar (45) | Not null |
| 36 | Penghasilan_ortu | Char (1) | Not null |
| 37 | Nama_wali | Varchar (45) | Not null |
| 38 | Tempat_lahir_wali | Varchar (150) | Not null |
| 39 | Tanggal_lahir_wali | Date | Not null |
| 40 | Pendidikan_wali | Varchar (1) | Not null |
| 41 | Pekerjaan_wali | Char (1) | Not null |
| 42 | Alamat_wali | Varchar (45) | Not null |
| 43 | Penghasilan_wali | Char (1) | Not null |

3. Tabel Migration

Primary Key : version

Foreign Key : -

Tabel 4.21 Migration

| No | Nama field | Tipe data | Keterangan |
|----|-----------------------|---------------|----------------|
| 1 | <u>Version</u> | Varvhar (180) | Auto_Increment |
| 2 | Apply_time | Int (11) | Not null |

4. Tabel Pembayaran

Primary Key : id_pembayaran_psb

Foreign Key : user_id, kasir

Tabel 4.22 Pembayaran

| No | Nama field | Tipe data | Keterangan |
|----|---------------------------------|--------------|----------------|
| 1 | <u>Id pembayaran psb</u> | Int (11) | Auto_Increment |
| 2 | Kasir | Int (11) | Not null |
| 3 | User_id | Int (11) | Not null |
| 4 | Tanggal_bayar | Datetime | Not null |
| 5 | Jumlah_bayar | Varchar (45) | Not null |
| 6 | Jenis_pembayaran | Char (1) | Not null |
| 7 | Nomorkwitansi | Char (12) | Not null |

5. Tabel Pengumuman

Primary Key : id_pengumuman

Foreign Key : id_admin

Tabel 4.23 Pengumuman

| No | Nama field | Tipe data | Keterangan |
|----|-----------------------------|---------------|----------------|
| 1 | <u>Id pengumuman</u> | Int (10) | Auto_Increment |
| 2 | Pengumuman | longtext (11) | Not null |
| 3 | Photo | Char (36) | Not null |

| | | | |
|---|---------------|---------------|----------|
| 4 | Create_at | Datetime | Not null |
| 5 | Id_admin | int (11) | Not null |
| 6 | Summary | Int (11) | Not null |
| 7 | Nomorkwitansi | varchar (200) | Not null |

6. Tabel *Setting Fitur*

Primary Key : id_setting

Foreign Key : -

Tabel 4.24 *Setting Fitur*

| No | Nama field | Tipe data | Keterangan |
|----|--------------------------|--------------|----------------|
| 1 | <u>Id_setting</u> | Int (10) | Auto_Increment |
| 2 | Setting | Varchar (45) | Not null |
| 3 | Status | Char (1) | Not null |

7. Tabel *setting value*

Primary Key : id_setting

Foreign Key : -

Tabel 4.25 *Setting Value*

| No | Nama field | Tipe data | Keterangan |
|----|--------------------------|---------------|----------------|
| 1 | <u>Id_setting</u> | Int (10) | Auto_Increment |
| 2 | Setting | Varchar (150) | Not null |
| 3 | Val | Varchar (45) | Not null |

8. Tabel User

Primary Key : id

Foreign Key : unit

Tabel 4.26 User

| No | Nama field | Tipe data | Keterangan |
|----|----------------------|---------------|----------------|
| 1 | <u>Id</u> | Int (12) | Auto_Increment |
| 2 | Username | Varchar (225) | Not null |
| 3 | Auth_key | Varchar (32) | Not null |
| 4 | Password_hash | Varchar (225) | Not null |
| 5 | Password_reset_token | Varchar (225) | Not null |
| 6. | Email | Varchar (225) | Not null |
| 7 | Status | Smallint (6) | Not null |
| 8 | Created_at | Date | Not null |
| 9 | Uupdate_at | Date | Not null |
| 10 | Unit | Tinyint (3) | Not null |

4.2.5 Implementasi Tampilan Halaman Sistem

Gambar dan penjelasan selanjutnya menunjukkan implementasi tampilan, akan memperlihatkan hasil halaman sistem yang telah dibuat :

a. Beranda

Terdapat tiga (menu) pada pojok kanan halaman yang dibuat yaitu menu beranda, daftar dan masuk. Menu beranda berisi tentang informasi mengenai pendaftaran siswa baru. Pada halaman menu Daftar user diarahkan untuk membuat akun sebelum mendaftar dengan mengisi username, email dan pasword. Sedangkan pada halaman

masuk atau login user harus mengisi username dan password yang sudah dibuat sebelumnya. Berikut ini tampilan dari halaman utama.



Gambar 4.17 Beranda

b. Halaman Daftar

Pada halaman daftar ini user diarahkan untuk membuat akun. Akun ini akan digunakan untuk login di aktivitas selanjutnya seperti merubah biodata, mencetak resi pembayaran, mencetak biodata dan melihat pengumuman.

Pendafataran

Username

Email

Password

Jenjang

▼

Daftar


Gambar 4.18 Halaman Daftar Akun

Setelah berhasil membuat akun langsung ke halaman pendaftaran. Pada proses pendaftaran ini semua data yang diinput akan tersimpan pada *database*.

| Informasi Siswa | Biodata Siswa | Biodata Orang Tua |
|----------------------------------------------------------------------------|-----------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| Photo <input type="button" value="Browse..."/> No file selected. | Nama Lengkap <input type="text"/> | Nama Ayah <input type="text"/> |
| NIK <input type="text"/> | Jenis Kelamin <input type="text" value="Pilih Jenis Kelamin..."/> | Pekerjaan Ayah <input type="text" value="Pilih Pekerjaan ayah..."/> |
| NISN <input type="text"/> | Agama <input type="text" value="Pilih Agama..."/> | Pendidikan Ayah <input type="text" value="Pilih Pendidikan Ayah..."/> |
| NO STTB <input type="text"/> | Tanggal Lahir <input type="text"/> | Tanggal Lahir Ayah <input type="text"/> |
| Tahun STTB <input type="text"/> | Tempat Lahir <input type="text"/> | Tempat Lahir Ayah <input type="text"/> |
| Nomor SKHUN <input type="text"/> | No Telepon <input type="text"/> | Nama Ibu <input type="text"/> |
| Nilai SKHUN <input type="text"/> | Gol Darah <input type="text" value="Pilih Golongan Darah..."/> | Pekerjaan Ibu <input type="text" value="Pilih Pekerjaan Ibu..."/> |
| Sekolah Asal <input type="text"/> | Anak Ke <input type="text"/> | Pendidikan Ibu <input type="text" value="Pilih Pendidikan Ibu..."/> |
| Alamat Sekolah Asal <input type="text"/> | Jumlah Saudara Kandung <input type="text"/> | Tanggal Lahir Ibu <input type="text"/> |

Gambar 4.19 Halaman Form Pendaftaran

Setelah mengisi *form* pendaftaran *user* harus mencetak resi untuk pembayaran psb sebesar Rp 150.000,- . berikut ini adalah cetak resi pembayaran :

|  YAYASAN PERGURUAN ISLAM AS SA'ADAH <small>Jalan Raya Kalimantan Sukarsa IB - 40 Pondok Kelapa Duren Sawit - Jakarta Timur 13450 Telp. 8640025</small> | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| Jenis Pembayaran : Pendaftaran Siswa Baru SMK | No Tagihan : 13202112015 |
| Nomor Peserta : 327100001 | |
| Nama Peserta : golden mangapul | |
| Jumlah Tagihan : Rp. 150.000 | |
| | Tanggal Akhir Pembayaran: Jun 30, 2015 |

Gambar 4.20 Cetak Resi Pembayaran

Setelah melakukan pendaftaran user harus mencetak biodata atau tanda bukti pendaftaran untuk keperluan validasi data. Berikut ini adalah tampilan cetak biodata :

Page: 1 of 1 Automatic Zoom

YAYASAN PERGURUAN ISLAM AS SA'ADAH
 Jl. Raya Kalimang Swakarsa I B No. 40 Pondok Kelapa
 Duren Sawit Jakarta Timur 13450 Telp. 8640025

Nomor Pendaftaran: 327100001

**FORMULIR PENDAFTARAN CALON SISWA SMK YPI AS SA'ADAH
 TAHUN PELAJARAN 2015**

A. Keterangan Calon Siswa

| | | | |
|-----|------------------------------|---|------------------------|
| 1. | Kelas | : | |
| 2. | Nama Siswa | : | golden mangapul |
| 3. | Jenis Kelamin | : | Pria |
| 4. | Tempat dan Tanggal Lahir | : | jakarta, 1990-02-06 |
| 5. | Alamat | : | bekasi Telp: 098789 |
| 6. | Agama | : | Protestan |
| 7. | Anak Ke | : | 1 |
| 8. | Jumlah Saudara Kandung | : | 2 |
| 9. | Kewarganegaraan | : | WNI |
| 10. | Anak Yatim/Piatu/Yatim Piatu | : | - |
| 11. | Nama Sekolah Asal | : | SMPN 10 Bekasi |
| 12. | Alamat Sekolah Asal | : | bekasi |
| 13. | Nomor STTB | : | 67667655gy |
| 14. | Nomor Induk Siswa Nasional | : | 6765456765 |
| 15. | Nomor Induk Kependudukan | : | 5454563636377776 |
| 16. | Nomor Peserta Ujian Nasional | : | 567656545656 |

B. Keterangan Kesehatan Siswa

| | | | |
|----|----------------|---|---|
| 1. | Golongan Darah | : | O |
|----|----------------|---|---|

Gambar 4.21 Cetak Biodata

c. Halaman Pengumuman

Pada halaman pengumuman berisi tentang pengumuman hasil seleksi. *User* bisa melihat pengumuman setelah *login*. Proses yang dilakukan pada halaman Pengumuman adalah menampilkan hasil seleksi diurutkan berdasarkan nilai terbesar ke nilai terkecil untuk SMK, SMP dan diurutkan berdasarkan umur untuk jenjang SD, TK. Berikut ini tampilan halaman seleksi:

Informasi Siswa

Menampilkan 1-2 dari 2 item.

| # | Nomor Peserta | Nama Lengkap | Sekolah Asal | Nilai SKHUN | Keterangan |
|---|----------------------|----------------------|----------------------|----------------------|------------|
| | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | |
| 1 | 327100001 | golden mangapul | SMPN 10 Bekasi | 80 | DITERIMA |
| 2 | 327100002 | cahyo pramudita | SMPN 2 Karawang | 70 | DITERIMA |

Gambar 4.22 Halaman Pengumuman Seleksi

d. Halaman *Login*

Pada halaman *login user* harus mempunyai akun terlebih dahulu, jika belum mempunyai akun harus membuat akun dihalaman *create akun*. Untuk *login User* mengisi *username* dan *password* dengan benar, jika benar akan berlanjut kehalaman biodata calon siswa, jika salah akan kembali ke halaman *login*. Berikut ini adalah tampilan dari *login user* :

Login

Username

Password

Login

Gambar 4.23 Form Login

e. Halaman *Login Admin*

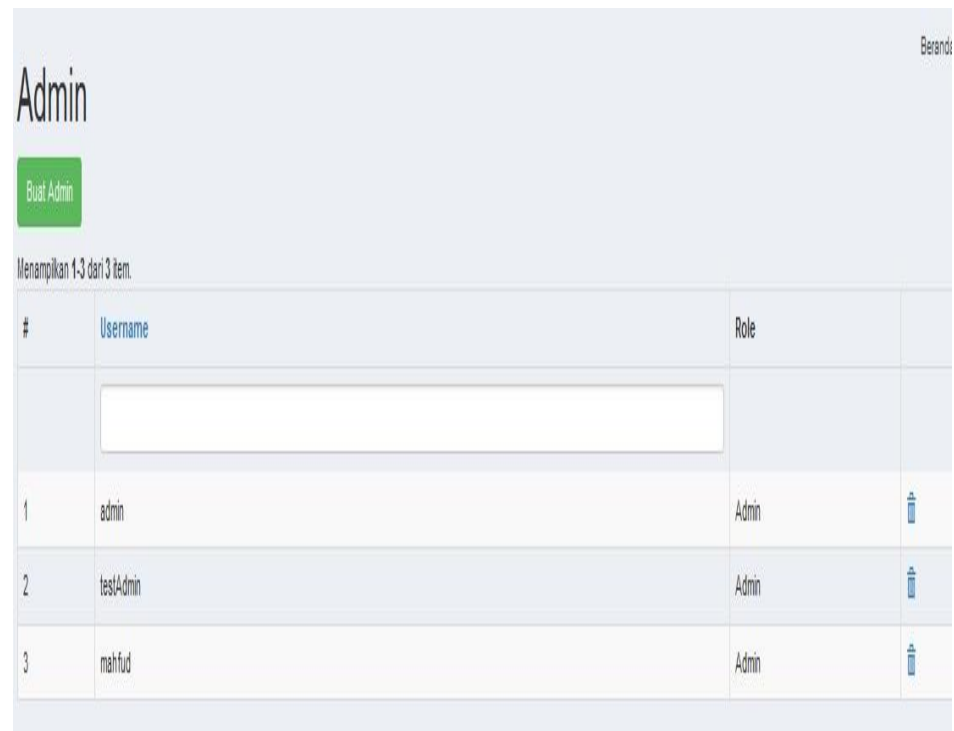
Pada halaman *login* admin masukan *username* dan *password*, jika *username* dan *password* benar akan masuk kehalaman dasbor admin. Jika *username* dan *password* salah akan kembali ke halaman *login*. Admin bertugas mengelola semua kegiatan yang ada di aplikasi penerimaan siswa baru, dari mulai pendaftaran sampai dengan seleksi. Berikut ini adalah tampilan dari *login* admin :



Gambar 4.24 *Form Login Admin*

f. Halaman Pendaftaran Admin

Pada halaman ini digunakan untuk membuat atau menghapus akun admin sesuai dengan kebutuhan pengelola. Berikut ini adalah tampilan dari halaman pendaftaran admin :



Gambar 4.25 Halaman Pendaftaran Admin





g. Halaman Pendaftar

Pada halaman pendaftar berisi calon siswa yang sudah melakukan input data. Pada halaman ini admin memiliki wewenang untuk mengupdate, melihat, menghapus dan registrasi data pendaftar. Untuk menu registrasi hanya bisa dilakukan ketika pendaftar sudah melakukan pembayaran. Prosedur registrasi disini yaitu pendaftar harus membayar PSB dan membawa berkas yang diperlukan sesuai dengan ketentuan yang berlaku. Berikut ini adalah halaman pendaftar pada admin :

Beranda > Calon Peserta

Calon Peserta

Menampilkan 1-2 dari 2 item.

| # | Nomor Peserta | Nama Lengkap | Jenis Kelamin | Agama | Tempat Lahir | Sekolah Asal | No Telepon | |
|---|---------------|---------------|---------------|-------|--------------|----------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | | | | | | | |
| 1 | 327100003 | bagus prabowo | Pria | Islam | jakarta | SMPI 8 jakarta | 034533 |   |
| 2 | 628100001 | hartato | Pria | Islam | jakarta | (belum diset) | 0657 |   |

Gambar 4.26 Calon Peserta pada Admin

Beranda > Calon Peserta

Update
Registrasi
Delete

| Informasi Siswa | Biodata Siswa | Biodata Orang Tua |
|-----------------------------------|--------------------------------------|-------------------------------------------|
| Photo My Application | Nama hartato | Nama Ayah hadi |
| Nomor Peserta 628100001 | Jenis Kelamin Pria | Pekerjaan Ayah PNS |
| NIK 3342452426242624 | Agama Islam | Pendidikan Ayah S1 |
| | Tanggal Lahir Dec 04, 2013 | Tempat Lahir Ayah jakarta |
| | Tempat Lahir jakarta | Tanggal Lahir Ayah Jul 11, 1990 |
| | Nomor Telepon 0657 | Nama Ibu rani |
| | Golongan Darah O | Pekerjaan Ibu Guru/Dosen |
| | Anak Ke- 2 | Pendidikan Ibu D3 |

Gambar 4.27 Form Registrasi

h. Halaman Calon Siswa

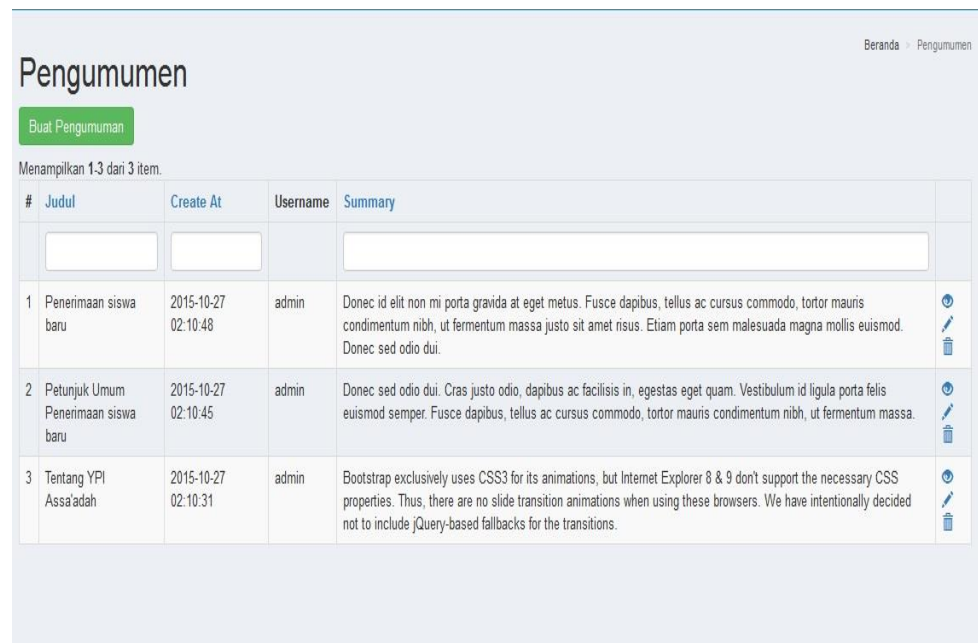
Pada halaman ini admin melihat semua calon siswa yang sudah membayar dan melakukan registrasi ke panitia penerimaan siswa baru. setelah pendaftaran sudah ditutup, admin dapat *export* data yang sudah valid dengan format *csv* yang berguna untuk laporan penerimaan siswa baru. Data juga dipakai untuk keperluan selanjutnya yaitu untuk data pembayaran sekolah, laporan siswa untuk ke dinas pendidikan terkait. Berikut ini adalah tampilan hasil calon siswa :

| # | Nomor Peserta | Nama Lengkap | Jenis Kelamin | Agama | Tempat Lahir | Sekolah Asal | No Telepon | |
|---|---------------|-----------------|---------------|-----------|--------------|-----------------|------------|--|
| 1 | 327100001 | golden mangapul | Pria | Protestan | jakarta | SMPN 10 Bekasi | 098789 | |
| 2 | 327100002 | cahyo pramudita | Pria | Islam | karawang | SMPN 2 Karawang | 021324 | |

Gambar 4.28 Tampilan Halaman Calon Siswa

g. Halaman Pengumuman Informasi

pada halaman ini admin mengelola mengenai informasi pendaftaran siswa baru, seperti alur pendafrtan dan lain – lain. Berikut ini adalah tampilan pengumuman pada admin:

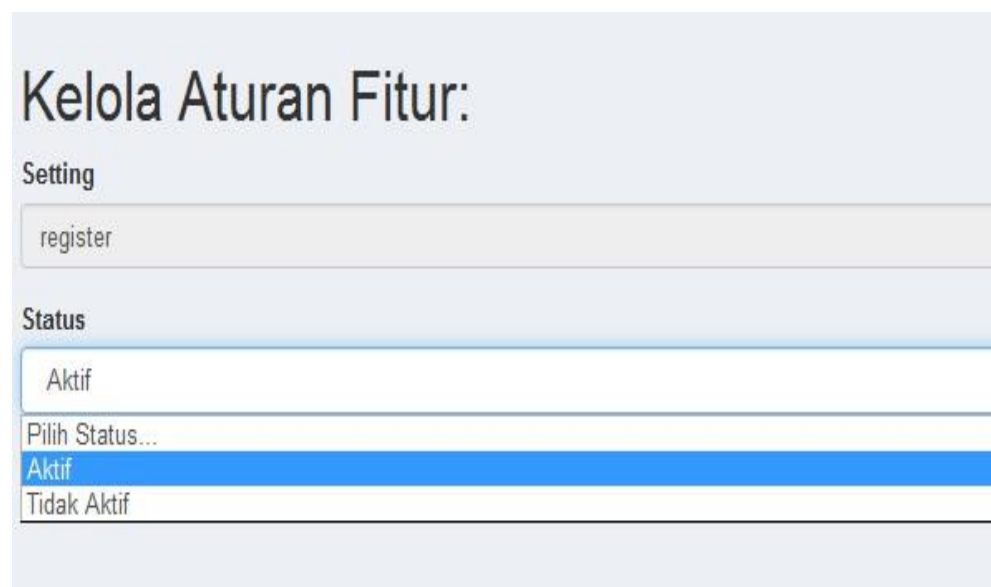


Gambar 4.29 Halaman Punguman Informasi

i. Halaman Atur Fitur Pendaftaran

Pada halaman ini berfungsi untuk mengatur dibuka di tutupnya pendaftaran siswa baru di YPI Assa'adah. Pada status aktif yaitu pendaftaran dibuka dan status tidak aktif yaitu pendaftaran di tutup.

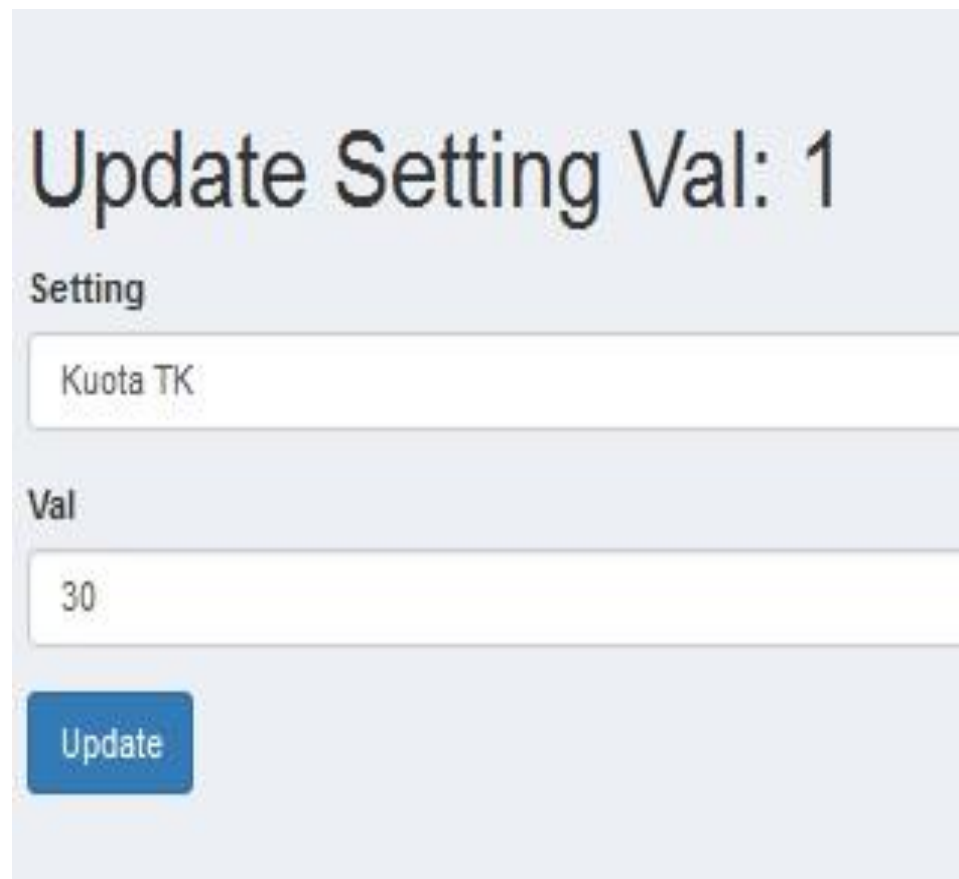
Berikut ini halaman atur fitur :



Gambar 4.30 Atur Fitur Pendaftaran

j. Setting Value Kuota

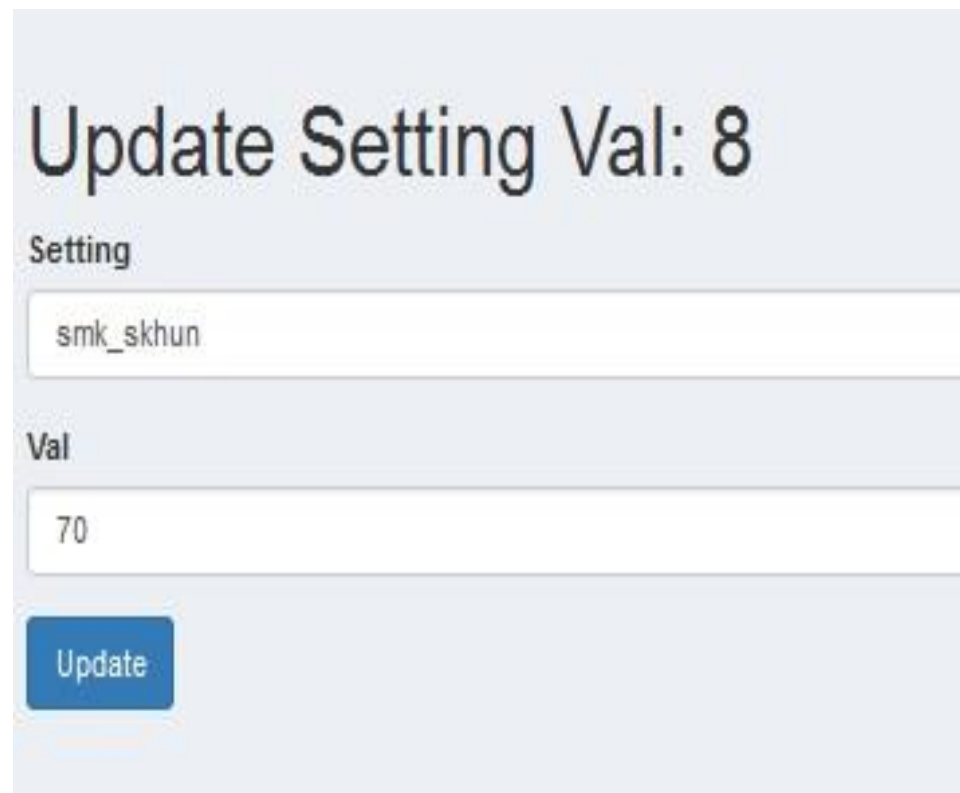
Pada halaman ini untuk mengatur jumlah kuota yang ditentukan oleh sekolah mengenai jumlah siswa yang akan diterima pada setiap jenjang yang ada. Berikut ini adalah contoh dari *setting value kuota*:



Gambar 4.31 Setting Value Kuota

k. Setting Value Nilai

Pada halaman ini untuk mengatur batas nilai terendah yang ditentukan oleh pihak sekolah. Penetapan nilai terendah ini guna untuk tahap penyeleksian pada jenjang SMK dan SMP. Sedangkan untuk jenjang TK dan SD yaitu dengan menentukan batas umur. Berikut ini adalah contoh dari *setting value nilai*:



Update Setting Val: 8

Setting

smk_skhun

Val

70

Update

Gambar 4.32 Setting Value Nilai

4.3 Pengujian

- 1) Pada penelitian ini, pengujian dilakukan dengan menggunakan metode *blackbox*. *Blackbox* adalah salah satu metode pengujian perangkat lunak yang berfokus pada sisi *fungsi*ionalitas, khususnya pada *input* dan *output* perangkat lunak. Fungsional yang akan di uji pada penelitian ini adalah fungsi dari semua menu yang telah dibuat, pada pengujian *blackbox*, menu pada *aplikasi* akan di uji satu persatu sesuai dengan kegunaan dan hasil yang diharapkan. Langkah awal tahap pengujian adalah menyediakan format pengujian *blackbox* . Format pengujian menggunakan metode *blackbox* dengan tipe *functional Testing* pada penelitian ini ditunjukkan berikut ini adalah hasil pengujian pada tabel 4.27 dan 4.28

Tabel 4.27 Hasil Pengujian Kebutuhan Fungsional User

| No | Fungsi | Skenario Proses | Sistem Bekerja | Hasil Proses | Keterangan |
|----|-------------|-------------------------------------------------------------------------|----------------|---------------------------------------|------------------------|
| 1 | Menu | User membuka halaman beranda, daftar, masuk/login | Ya | Berita yang tersedia muncul dihalaman | Benar |
| 2 | Create akun | User mengisi username, email, dan password pada form daftar | Ya | User berhasil membuat akun | Benar |
| 3 | Create akun | User mengisi username saja pada form create akun | Ya | User gagal membuat akun | Muncul pesan kesalahan |
| 4 | Create akun | User mengisi username dan email saja pada form create akun | Ya | User gagal membuat akun | Muncul pesan kesalahan |
| 5 | Login | User login menggunakan username dan password yang benar | Ya | User berhasil login | Benar |
| 6 | Login | User melakukan login dengan username yang benar dan password yang salah | Ya | User gagal login | Muncul pesan kesalahan |
| 7 | Login | User melakukan login dengan username yang salah dan password yang benar | Ya | User gagal login | Muncul pesan kesalahan |

| | | | | | |
|----|--------------------------|------------------------------------------------------------------------------------|----|--------------------------------------------|------------------------|
| 8 | Login | User melakukan login dengan username yang salah dan password yang salah | Ya | User gagal login | Muncul pesan kesalahan |
| 9 | Lupa password | User lupa password dengan mengklik lupa password | Ya | User menerima konfirmasi melalui email | Muncul notifikasi |
| 10 | Form pendaftaran | User mengisi dengan benar semua form pendaftaran yang disediakan lalu tekan simpan | Ya | User berhasil mengisi data diri | Benar |
| 11 | Form pendaftaran | User tidak mengisi bagian penting yang ada di form pendaftaran lalu tekan simpan | Ya | User gagal mengisi daftar diri | Muncul pesan kesalahan |
| 12 | Perbaharui data | user memperbaharui data diri yang sudah tersimpan | Ya | User berhasil memperbaharui data diri | Benar |
| 13 | Mencetak resi pembayaran | User mencetak resi pembayaran | Ya | User berhasil menyimpan atau mencetak resi | Benar |
| 14 | Mencetak biodata | User mencetak biodata yang sudah tersimpan | Ya | User berhasil mencetak bukti pendaftaran | Benar |
| 15 | Logout | User menekan menu logout | Ya | User berhasil logout atau keluar | Benar |

Tabel 4.28 Hasil Pengujian Kebutuhan Fungsional Admin

| No | Fungsi | Skenario Proses | Sistem Bekerja | Hasil Proses | Keterangan |
|----|-------------------------|--------------------------------------------------------------------------|----------------|------------------------------------------|------------------------|
| 1 | Login | admin login menggunakan username dan password yang benar | Ya | Admin berhasil login | Benar |
| 2 | Login | admin melakukan login dengan username yang benar dan password yang salah | Ya | Admin gagal login | Muncul pesan kesalahan |
| 3 | Login | admin melakukan login dengan username yang salah dan password yang benar | Ya | Admin gagal login | Muncul pesan kesalahan |
| 4 | Login | Admin melakukan login dengan username yang salah dan password yang salah | Ya | Admin gagal login | Muncul pesan kesalahan |
| 5 | Kelola Admin | Admin menambahkan akun baru dan menghapus akun | Ya | Admin berhasil menambahkan akun baru | Benar |
| 6 | Pendaftaran | Admin melihat pendaftar yang sudah masuk | Ya | Admin melihat pendaftar yang sudah masuk | Benar |
| 7 | Kelola data calon siswa | Admin mengupdate data calon siswa jika diperlukan dengan mengklik update | Ya | Admin mengupdate data calon siswa | Benar |

| | | | | | |
|----|-------------------------|-------------------------------------------------------------------------------------|----|-------------------------------------------------------------|-------|
| 8 | Kelola data calon siswa | Admin mendelete data calon siswa dengan mengklik delete | Ya | Admin berhasil menggunakan perubahan data | Benar |
| 9 | Kelola data calon siswa | Admin melakukan registrasi pendaftar menjadi calon siswa dengan mengklik registrasi | Ya | Admin berhasil meregistrasi calon siswa baru | Benar |
| 10 | Kelola data calon siswa | Admin dapat merubah kembali status calon siswa menjadi pendaftar | Ya | Admin berhasil merubah status calon siswa baru | Benar |
| 11 | Kelola data calon siswa | Admin dapat melihat daftar calon siswa yang sudah falid di menu calon siswa | Ya | Admin berhasil melihat data yg sudah falid | Benar |
| 12 | Kelola data calon siswa | Admin dapat mengexport data calon siswa yang sudah falid kefile exel | Ya | Admin berhasil mengexport data kedalam format csv atau exel | Benar |
| 13 | Kelola kelas | Admin dapat melakukan pembagian kelas | Ya | Informasi akan muncul dihalaman laporan | Benar |
| 14 | Kelola data calon siswa | Admin dapat menghapus data calon siswa di menu calon siswa | Ya | Data tersimpan di database | Benar |

| | | | | | |
|----|-------------------------------|------------------------------------------------------------------------------------|----|--------------------------------------|-------|
| 15 | Kelola data calon siswa | Admin dapat mengupdate data calon siswa di menu calon siswa | Ya | Data tersimpan di database | Benar |
| 16 | Kelola data calon siswa | admin dapat membuat pengumuman tentang penerimaan siswa baru | Ya | Informasi akan muncul dihalaman | Benar |
| 17 | Kelola pengumuman atau berita | Admin dapat mengupdate pengumuman | Ya | Informasi akan muncul di halaman | Benar |
| 18 | Kelola pengumuman atau berita | Admin dapat menghapus pengumuman | Ya | Informasi akan tersimpan di database | Benar |
| 19 | Kelola fitur pendaftaran | Admin dapat mengatur fitur pendaftaran | Ya | Informasi tampil di halaman | Benar |
| 20 | Kelola kuota | Admin dapat menentukan kuota calon siswa di setiap jenjang | Ya | Perubahan tersimpan di database | Benar |
| 21 | Kelola kuota | Admin dapat mengubah kuota di setiap jenjang | Ya | Perubahan tersimpan di database | Benar |
| 22 | Kelola seleksi | Admin dapat menentukan batas minimum nilai pada proses seleksi jenjang SMP dan SMK | Ya | Perubahan tersimpan di database | Benar |

| | | | | | |
|----|----------------|--------------------------------------------------------------------------------------|----|---------------------------------|-------|
| 23 | Kelola seleksi | Admin dapat merubah batas nilai minimum pada proses seleksi pada jenjang SMP dan SMK | Ya | Perubahan tersimpan di database | Benar |
| 24 | Kelola seleksi | Admin dapat menentukan batas umur minimum pada seleksi jenjang TK dan SD | Ya | Perubahan tersimpan di database | Benar |
| 25 | Kelola seleksi | Admin dapat merubah batas umur minimum pada seleksi jenjang TK dan SMP | Ya | Perubahan tersimpan di database | Benar |
| 26 | Lihat grafik | Admin dapat melihat grafik pendatar yang sudah masuk | Ya | Inforasi akan muncul di halaman | Benar |
| 27 | Logout | Admin mengklik Logout | Ya | Admin berhasil logout | Benar |

Pada peneltian ini, penguji mengambil data sample untuk melakukan pengujian dengan mengambil 15 siswa dari jenjang TK,SD, SMP dan SMK siswa baru pada tahun 2015 – 2016 di YPI Assa’adah. Tabel skenario pengujian adalah tabel yang berisi instrumen yang akan diuji, instrumen yang di uji pada format ini adalah pengujian dari segi fungsi perangkat lunak yang dibuat. Berdasarkan pengujian diatas fungsi sudah sesuai dengan kebutuhan yang ada di YPI Assa’adah. Aplikasi sudah bekerja dan sesuai dengan kebutuhan, maka aplikasi dikatakan berhasil

4.4 Pemeliharaan

Pemeliharaan dianggap perlu dilakukan untuk memperbaiki jika ada kesalahan sistem yang tidak berjalan sesuai dengan fungsinya yang mungkin disebabkan karena perubahan tampilan atau perubahan kode program. Walaupun sudah diuji sebelumnya. Jika belum sesuai maka akan kembali ketahap awal. Jika sudah sesuai dengan permintaan maka pemeliharaan cukup sampai disini. Pada aplikasi penerimaan siswa baru ini tidak ada perbaikan karena analisis kebutuhannya sudah jelas dari awal pembuatan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pengembangan Aplikasi penerimaan siswa baru berbasis web merupakan salah satu solusi yang tepat bagi Yayasan Pendidikan Islam As Sa'adah. dari proses dan hasil penelitian, peneliti dapat mengambil kesimpulan :

1. Berdasarkan pengujian aplikasi penerimaan siswa baru pada YPI Asa'adah sudah sesuai dengan tujuan penelitian.
2. Aplikasi penerimaan siswa baru yang dikembangkan mampu membantu tugas pokok dalam melakukan proses penyeleksian, penyimpanan data dan pengolahan data sesuai dengan kebutuhan

5.2 Saran

Sistem informasi penerimaan siswa baru yang telah dikembangkan ini memiliki beberapa kekurangan, sehingga perlu adanya pengembangan dalam menyempurnakan aplikasi ini. Berikut ini beberapa kekurangan yang harus dikembangkan :

1. Dalam sistem penerimaan siswa baru ini terdapat banyak kelemahan terutama keamanan (*security*) sistem. Peneliti berharap pada penelitian selanjutnya masalah keamanan sistem harus diperbaiki.

2. Adanya pembayaran melalui banking atau ATM agar memudahkan siswa dalam melakukan pembayaran pendaftaran agar dapat memudahkan calon siswa dalam melakukan pembayaran.
3. Untuk penelitian selanjutnya diharapkan dapat menambahkan fasilitas pendukung.

DAFTAR PUSTAKA

- Abdul Kadir. 2009. *Dasar dan Perancangan dan Implementasi Database Relasional*. Yogyakarta : Andi.
- Bauer C, King G. 2007. *Java Persistence with Hibernate*. United States :Manning.
- Bonczek, R. H.; Holsapple, C. W.; & Whinston, A. B. 1980, *The Evolving Roles of Models in Decision Support Systems*, Decision Science, Vol. 11, No. 2.
- Fakultas Teknik. 2012. Buku Pedoman Skripsi / Komprehensif / Karya Inovatif (S1). Jakarta, UNJ cetakan II
- Hanna, "Farewall to waterfalls", software magazine, Mei 1995
- Hendrayudi, VB. 2008. *Untuk Berbagai Keperluan Pemrograman*. Jakarta: PT. Elex Media Komputindo
- Hengky W. Pranama.2012. *Kunci Sukses Aplikasi Penjualan Berbasis Access*. Jakarta: AlexMedia Komputerindo.
- Husted T. 2003. *Struts in Action : Building Web Application with the Leading Java Framework*. United States : Manning.
- Indrajani. 2014. *Pengantar Sistem Basis Data*. Jakarta: PT Elex Media Komputindo
- Janner Simartama. 2010. *Rekayasa Perangkat Lunak*. Yogyakarta andi
- Kamus Besar Bahasa Indonesia , Edisi keempat. 2008. Jakarta: Gramedia Pusaka Utama
- Nur widiyanto. 2010. *Membangun Aplikasi Java Enterprise dengan Arsitektur Model View Controller*. Andi publisher
- Roger Pressman.2007. *Rekayasa Perangkat Lunak (buku satu)*. Yogyakarta andi
- Rosa, & M. Shalaludin. 2014. *Rekayasa Perangkat Lunak*. Bandung : Informatika.
- Yuhefizar. 2012. *10 Jam Menguasai Teknologi dan Aplikasinya*. Jakarta: Alex Media Koputerindo.
- YM Kusuma. 2012. *PHP Menyelesaikan Website 30 juta*. Jakarta: jaskom.

Lampiran 01

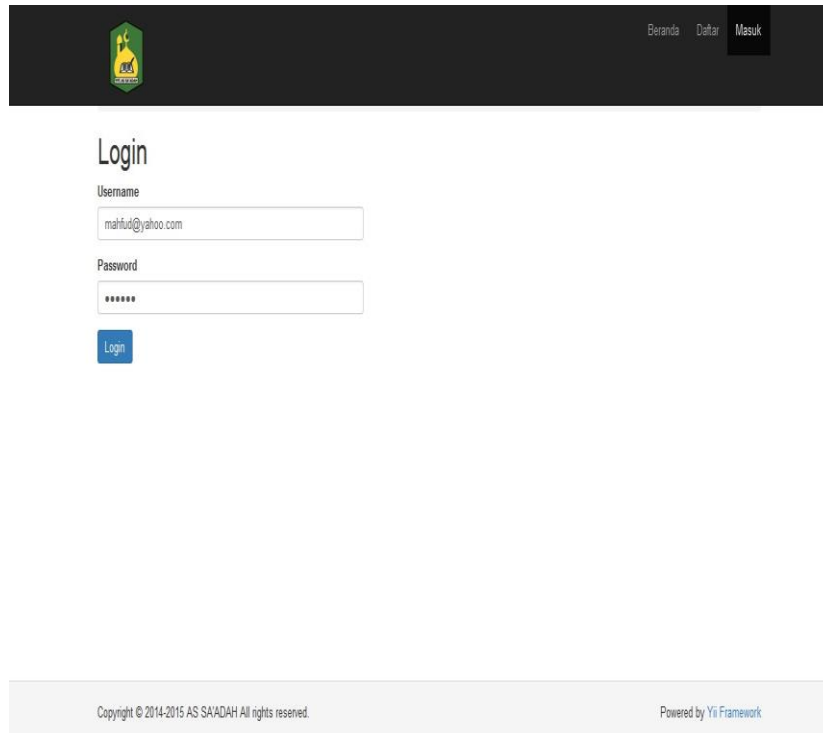
Hasil tampilan program

1. Beranda



2. Daftar atau Create akun

3. Login



[Beranda](#)
[Daftar](#)
[Masuk](#)

Login

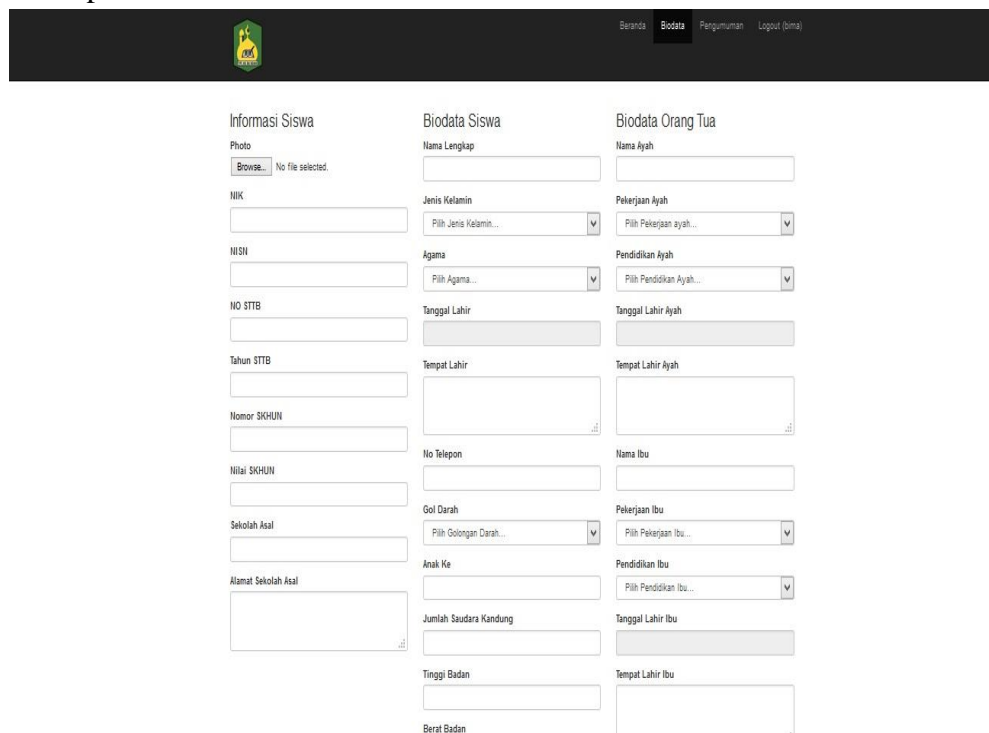
Username

Password

[Login](#)

Copyright © 2014-2015 AS SAADAH All rights reserved. [Powered by Yi Framework](#)

4. Form pendaftaran




[Beranda](#)
[Biodata](#)
[Pengumuman](#)
[Logout \(bima\)](#)

Form pendaftaran

| Informasi Siswa | Biodata Siswa | Biodata Orang Tua |
|---------------------------------------------------------|----------------------------------------------------------------------|--------------------------------------------------------------------------|
| Photo <input type="text" value="No file selected."/> | Nama Lengkap <input type="text"/> | Nama Ayah <input type="text"/> |
| NIK <input type="text"/> | Jenis Kelamin <input type="text" value="Pilih Jenis Kelamin..."/> | Pekerjaan Ayah <input type="text" value="Pilih Pekerjaan ayah..."/> |
| NISN <input type="text"/> | Agama <input type="text" value="Pilih Agama..."/> | Pendidikan Ayah <input type="text" value="Pilih Pendidikan Ayah..."/> |
| NO STTB <input type="text"/> | Tanggal Lahir <input type="text"/> | Tanggal Lahir Ayah <input type="text"/> |
| Tahun STTB <input type="text"/> | Tempat Lahir <input type="text"/> | Tempat Lahir Ayah <input type="text"/> |
| Nomor SKHUN <input type="text"/> | No Telepon <input type="text"/> | Nama Ibu <input type="text"/> |
| Nilai SKHUN <input type="text"/> | Gol Darah <input type="text" value="Pilih Golongan Darah..."/> | Pekerjaan Ibu <input type="text" value="Pilih Pekerjaan Ibu..."/> |
| Sekolah Asal <input type="text"/> | Anak Ke <input type="text"/> | Pendidikan Ibu <input type="text" value="Pilih Pendidikan Ibu..."/> |
| Alamat Sekolah Asal <input type="text"/> | Jumlah Saudara Kandung <input type="text"/> | Tanggal Lahir Ibu <input type="text"/> |
| | Tinggi Badan <input type="text"/> | Tempat Lahir Ibu <input type="text"/> |
| | Berat Badan <input type="text"/> | |

5. Cetak resi pembayaran

Page: 1 of 1 Automatic Zoom




**YAYASAN PERGURUAN ISLAM
AS SA'ADAH**
Jalan Raya Kalimang Swakarsa I B No. 40 Pondok Kelapa
Duren Sawit - Jakarta Timur 13450
Telp. 8640025

| | | | |
|------------------|------------------------------|------------|---------------|
| Jenis Pembayaran | : Pendaftaran Siswa Baru SMK | No Tagihan | : 13202112015 |
| Nomor Peserta | : 327100001 | | |
| Nama Peserta | : golden mangapul | | |
| Jumlah Tagihan | : Rp. 150.000 | | |

Tanggal Akhir Pembayaran:
Jun 30, 2015

6. Cetak biodata

Page: 1 of 1 Automatic Zoom



**YAYASAN PERGURUAN ISLAM
AS SA'ADAH**
Jl. Raya Kalimang Swakarsa I B No. 40 Pondok Kelapa
Duren Sawit Jakarta Timur 13450 Telp. 8640025


Nomor Pendaftaran: 327100001

**FORMULIR PENDAFTARAN CALON SISWA SMK YPI AS SA'ADAH
TAHUN PELAJARAN 2015**

| | | | |
|--------------------------------------|------------------------------|---|---------------------|
| A. Keterangan Calon Siswa | | | |
| 1. | Kelas | : | |
| 2. | Nama Siswa | : | golden mangapul |
| 3. | Jenis Kelamin | : | Pria |
| 4. | Tempat dan Tanggal Lahir | : | jakarta, 1990-02-06 |
| 5. | Alamat | : | bekasi |
| 6. | Agama | : | Telp: 098789 |
| 7. | Anak Ke | : | Protestan |
| 8. | Jumlah Saudara Kandung | : | 1 |
| 9. | Kewarganegaraan | : | 2 |
| 10. | Anak Yatim/Piatu/Yatim Piatu | : | WNI |
| 11. | Nama Sekolah Asal | : | - |
| 12. | Alamat Sekolah Asal | : | SMPN 10 Bekasi |
| 13. | Nomor STTB | : | bekasi |
| 14. | Nomor Induk Siswa Nasional | : | 67667655gy |
| 15. | Nomor Induk Kependudukan | : | 6765456765 |
| 16. | Nomor Peserta Ujian Nasional | : | 54545636377776 |
| B. Keterangan Kesehatan Siswa | | | |
| 1. | Golongan Darah | : | 567656545656 |
| | | : | O |

uitshub-aakamaihd.net...

7. Pengumuman


[Beranda](#)
[Biodata](#)
[Pengumuman](#)
[Logout \(bima\)](#)

Informasi Siswa

Menampilkan 1-3 dari 3 item.

| # | Nomor Peserta | Nama Lengkap | Sekolah Asal | Nilai SKHUN | Keterangan |
|---|----------------------|----------------------|----------------------|----------------------|------------|
| | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | |
| 1 | 327100003 | bagus prabowo | SMPN 8 jakarta | 90 | LULUS |
| 2 | 327100001 | golden mangapul | SMPN 10 Bekasi | 80 | LULUS |
| 3 | 327100002 | cahyo pramudita | SMPN 2 Karawang | 70 | LULUS |

Copyright © 2014-2015 AS SAADAH All rights reserved.

Powered by [Yii Framework](#)

8. Login admin

Login

Username

Password

9. tambah admin

Admin Pendaftaran ≡ Beranda About Logout (admin)

Pendaftaran Pengumuman Fitur Setting Value Grafik

Admin

[Buat Admin](#)

Menampilkan 1-3 dari 3 item.

| # | Username | Role | |
|---|-----------|-------|--|
| 1 | admin | Admin | |
| 2 | testAdmin | Admin | |
| 3 | mahfud | Admin | |

10. Pendaftar pada admin

Admin Pendaftaran ≡ Beranda About Logout (admin)

Pendaftaran Pengumuman Fitur Setting Value Grafik

Calon Peserta

Menampilkan 1-2 dari 2 item.

| # | Nomor Peserta | Nama Lengkap | Jenis Kelamin | Agama | Tempat Lahir | Sekolah Asal | No Telepon | |
|---|---------------|---------------|---------------|-------|--------------|----------------|------------|--|
| 1 | 327100003 | bagus prabowo | Pria | Islam | jakarta | SMPN 8 jakarta | 034533 | |
| 2 | 628100001 | hartato | Pria | Islam | jakarta | (belum diset) | 0657 | |

11. Calon siswa pada admin

Admin Pendaftaran Beranda About Logout (admin)

Calon Siswa Beranda > Calon Siswa

[Export Calon Siswa](#)

Menampilkan 1-2 dari 2 item.

| # | Nomor Peserta | Nama Lengkap | Jenis Kelamin | Agama | Tempat Lahir | Sekolah Asal | No Telepon |
|---|---------------|-----------------|---------------|-----------|--------------|-----------------|------------|
| 1 | 327100001 | golden mangapul | Pria | Protestan | jakarta | SMPN 10 Bekasi | 098789 |
| 2 | 327100002 | cahyo pramudita | Pria | Islam | karawang | SMPN 2 Karawang | 021324 |

12. Pengumuman pada admin

Admin Pendaftaran Beranda About Logout (admin)

Pengumuman Beranda > Pengumuman

[Buat Pengumuman](#)

Menampilkan 1-3 dari 3 item.

| # | Judul | Create At | Username | Summary |
|---|-------------------------------------|---------------------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Penerimaan siswa baru | 2015-10-27 02:10:48 | admin | Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui. |
| 2 | Petunjuk Umum Penerimaan siswa baru | 2015-10-27 02:10:45 | admin | Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa. |
| 3 | Tentang YPI Assa'adah | 2015-10-27 02:10:31 | admin | Bootstrap exclusively uses CSS3 for its animations, but Internet Explorer 8 & 9 don't support the necessary CSS properties. Thus, there are no slide transition animations when using these browsers. We have intentionally decided not to include jQuery-based fallbacks for the transitions. |

13. Fitur pada admin

Admin Pendaftaran

Beranda About Logout (admin)

Pendaftaran <

Pengumuman

Fitur

Setting Value

Grafik <

Beranda > Kelola Aturan > 1 > Update

Kelola Aturan Fitur:

Setting

register

Status

Aktif

Update

14. Setting value pada admin

Admin Pendaftaran

Beranda About Logout (admin)

Pendaftaran <

Pengumuman

Fitur

Setting Value

Grafik <

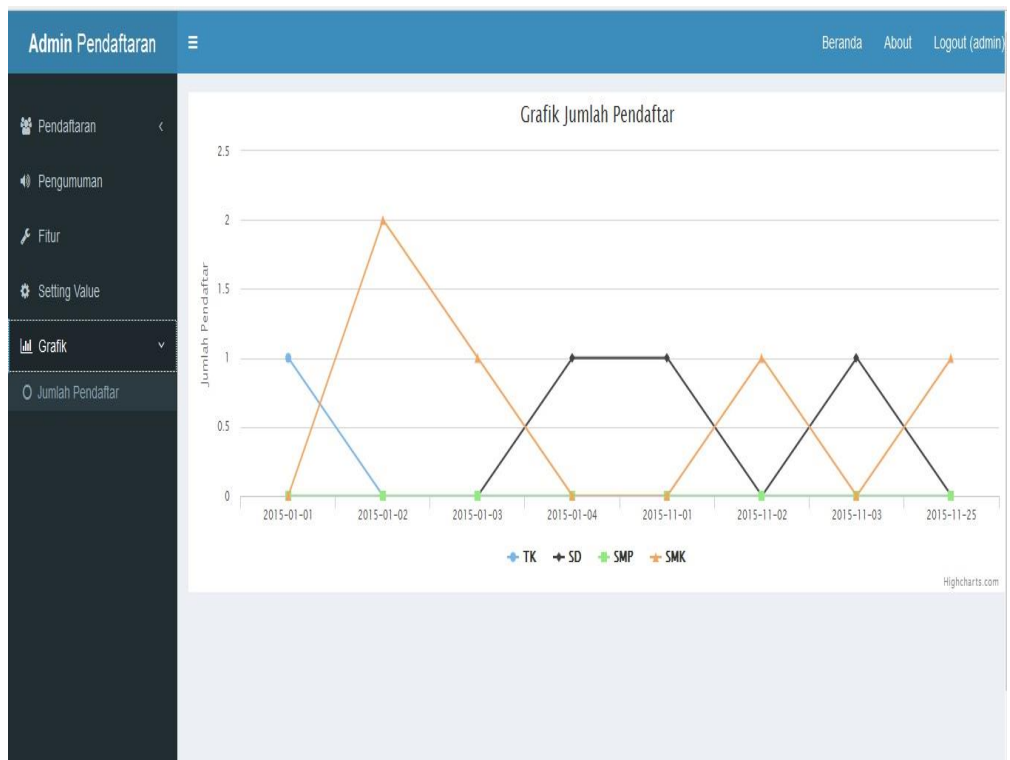
Beranda > Setting Value

Setting Value

Menampilkan 1-8 dari 8 item.

| # | Setting | Val | |
|---|----------------------|----------------------|--|
| | <input type="text"/> | <input type="text"/> | |
| 1 | Kuota TK | 30 | |
| 2 | Kuota SD | 30 | |
| 3 | Kuota SMP | 30 | |
| 4 | Kuota SMK | 30 | |
| 5 | tk_age | 4 | |
| 6 | sd_age | 7 | |
| 7 | smp_skhun | 30 | |
| 8 | smk_skhun | 70 | |

15. Grafik pada admin



Lampiran 02



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS NEGERI JAKARTA

Kampus Universitas Negeri Jakarta, Jalan Rawamangun Muka, Jakarta 13220
Telp./Fax. : Rektor : (021) 4893854, PR I : 4895130, PR II : 4893918, PR III : 4892926, PR IV : 4893982
BAUK : 4750930, BAAK : 4759081, BAPSI : 4752180
Bag. UHTP : Telp. 4893726, Bag. Keuangan : 4892414, Bag. Kepegawaian : 4890536, HUMAS : 4898486
Laman : www.unj.ac.id

Nomor : 1439/UN39.12/KM/2015
Lamp. : -
Hal : Permohonan Izin Mengadakan Penelitian
untuk Penulisan Skripsi

31 Maret 2015

Yth. Ketua YPI As'Saadah Pondok Kelapa
Jakarta Timur

Kami mohon kesediaan Saudara untuk dapat menerima Mahasiswa Universitas Negeri Jakarta :

Nama : Mahfud Ramdhani A.
Nomor Registrasi : 5235107392
Program Studi : Pendidikan Teknik Informatika dan Komputer
Fakultas : Teknik Universitas Negeri Jakarta
No. Telp/HP : 08999840404

Dengan ini kami mohon diberikan ijin mahasiswa tersebut, untuk dapat mengadakan penelitian guna mendapatkan data yang diperlukan dalam rangka Penulisan Skripsi. Skripsi tersebut dengan judul :

"Pengembangan Aplikasi Penerimaan Siswa Baru Pada Yayasan Perguruan Islam As'Saadah Pondok Kelapa"

Atas perhatian dan kerjasama Saudara, kami sampaikan terima kasih.

Kepala Biro Administrasi
Akademik dan Kemahasiswaan,



Tembusan :
1. Dekan Fakultas Teknik
2. Kaprog / Jurusan Teknik Elektro

Drs. Syaifullah
NP 195702161984031001



YAYASAN PERGURUAN ISLAM AS SA'ADAH

Jl. Raya Kalimalang Swakarsa I B No. 40 - 004/03 Pondok Kelapa
Kecamatan Duren Sawit - Jakarta Timur 13450 Telp/Fax. (021) 8640025

SURAT KETERANGAN

Nomor: 306/ YPIA/ IV/ 2015

Menindaklanjuti permohonan izin mengadakan penelitian untuk penulisan skripsi di Yayasan Perguruan Islam As Sa'adah Pondok Kelapa tertanggal 31 Maret 2015, yang bertanda tangan dibawah ini Ketua Yayasan Perguruan Islam As Sa'adah Pondok Kelapa.

Menerangkan bahwa

Nama : Mahfud Ramdhani A

Nomor Registrasi : 5235107392

Program Studi : Pendidikan Teknik Informatika dan Komputer

Dengan ini kami memberikan ijin mahasiswa tersebut, untuk dapat mengadakan penelitian guna mendapatkan data yang diperlukan dalam rangka Penulisan Skripsi. Skripsi tersebut dengan judul :

"Pengembangan Aplikasi Penerimaan Siswa Baru Pada Yayasan Perguruan Islam As Sa'adah Pondok Kelapa"

Demikian surat keterangan ini dibuat dengan sebenarnya untuk dapat dipergunakan sebagai mana mestinya.

Ketua,

Yayasan Perguruan Islam As Sa'adah
Pondok Kelapa



Drs. Munier Arsyad, M.Pd.



YAYASAN PERGURUAN ISLAM AS SA'ADAH

Jl. Raya Kallimalang Swakarsa I B No. 40 - 004/03 Pondok Kelapa
Kecamatan Duren Sawit - Jakarta Timur 13450 Telp/Fax. (021) 8640025

LEMBAR PENGESAHAN

Nomor: 363/ YPIA/ IX/ 2015

Yang bertanda tangan di bawah ini Ketua Yayasan Perguruan Islam As Sa'adah Pondok Kelapa, Mengesahkan penelitian untuk penulisan skripsi yang berjudul :

"Pengembangan Aplikasi Penerimaan Siswa Baru Pada Yayasan Perguruan Islam As Sa'adah Pondok Kelapa".

Penelitian tersebut dilaksanakan oleh :

Nama : Mahfud Ramdhani A

Nomor Registrasi : 5235107392

Program Studi : Pendidikan Teknik Informatika dan Komputer

Demikian surat pengesahan ini dibuat dengan sebenarnya untuk dapat dipergunakan sebagai mana mestinya.

Disahkan : Di Jakarta

Tanggal : 30 September 2015



Ketua,
Yayasan Perguruan Islam As Sa'adah
Pondok Kelapa

Drs. Munier Arsyad, M.Pd.



YAYASAN PERGURUAN ISLAM AS SA'ADAH

Jl. Raya Kalimalang Swakarsa I B No. 40 Pondok Kelapa
Duren Sawit Jakarta Timur 13450 Telp. 8640025

Nomor Pendaftaran : 0088 *

FORMULIR PENDAFTARAN CALON SISWA YPI AS SA'ADAH TAHUN PELAJARAN

A. Keterangan Calon siswa

1. Kelas :
2. Nama Siswa :
3. Jenis Kelamin : ☐ [L] Pria ☐ [P] Wanita :
4. Tempat dan Tanggal Lahir :
5. Alamat : Jl. Tlp.
6. Agama :
7. Anak ke :
8. Jumlah saudara kandung :
9. Kewarganegaraan :
10. Anak yatim/piatu/yatim piatu :
11. Nama Sekolah Asal :
12. Alamat Sekolah Asal :
13. Nomor STTB dan tahun : Nomor Tahun
14. Nomor Induk Siswa Nasional :
15. Nomor Induk Kependudukan :

PAS FOTO
3 X 4

B. Keterangan Kesehatan Siswa

1. Golongan Darah :
2. Penyakit yang pernah diderita :
3. Kelainan Jasmani :
4. Tinggi dan berat badan : Cm/ Kg
5. Jarak tempat tinggal ke sekolah : Km/m

C. Keterangan Orang Tua/Wali Siswa

1. Nama Lengkap a. Ayah :
b. Ibu :
2. Tempat tanggal lahir a. Ayah :
b. Ibu :
3. Pendidikan tertinggi a. Ayah : ☐ [A] SD/MI/Paket A ☐ [B] SMP/MTs/Paket B ☐ [C] SMA/MA/Paket C ☐ [D] Diploma 1
b. Ibu : ☐ [E] Diploma 2 ☐ [F] Diploma 3 ☐ [G] S-1 ☐ [H] S-2 [Magister]
☐ [I] S-3 [Doktor]
4. Pekerjaan a. Ayah : ☐ [A] Petani ☐ [B] PNS ☐ [C] ABRI ☐ [D] Guru/Dosen
b. Ibu : ☐ [E] Dokter ☐ [F] Politikus ☐ [G] Pedagang / Wiraswasta ☐ [H] Pegawai Swasta
☐ [I] Buruh ☐ [J] Seni/Lukis/Artis/Sejenis ☐ [K] Dirumah ☐ [L] Lain-lain
5. Penghasilan Orang Tua : ☐ [A] < 500 rb ☐ [B] 500 rb - 1 jt ☐ [C] 1 - 3 jt ☐ [D] 3 - 5 jt ☐ [E] > 5 jt
6. Alamat Orang Tua :

Petugas Pendaftaran

Orang tua / wali

Jakarta,

Calon siswa

Nama Jelas

Nama Jelas

Nama Jelas

*) Di isi oleh petugas pendaftaran

TK ISLAM / SDI / SMP / SMK PLUS

Lampiran 03

WAWANCARA

Hari/Tanggal : Rabu . 01 APRIL 2015

Nama : DJOKO TRIANTO

Jabatan : KETUA PSB YPI AS SA'ADAH

Tempat : YPI AS SA'ADAH

Pertanyaan :

1. Bagaimana proses pendaftaran siswa baru yang sudah berjalan di YPI Assa'adah saat ini?
2. Apakah ada rencana untuk membuat sistem pendaftaran siswa baru yang terkomputerisasi agar mempermudah proses penerimaan siswa baru?
3. Apakah ada masalah-masalah didalam proses penerimaan siswa baru?
4. Apa saja persyaratan calon siswa yang diperlukan untuk melakukan pendaftaran?
5. Bagaimana proses seleksi penerimaan siswa baru yang ada di YPI Assa'adah?

Jawaban :

1. Dalam melakukan proses pendaftaran di YPI Assa'addah yaitu :
 - a. Pihak Sekolah membagikan Selebaran Formulir tentang Seputar YPI Assa'addah khususnya di daerah Pondok Kelapa, Jakarta
 - b. Menunggu kedatangan calon pendaftar yang datang langsung ke Sekolah
 - c. Calon siswa mengisi Formulir Pendaftaran dan membayar formulir Pendaftaran. Setelah formulir di isi dengan benar langsung di Serahkan kepanitia.
 - d. Calon siswa menyerahkan berkas yang di perlukan untuk validasi dan daftar ulang.
2. Ada rencana untuk membuat sistem Pendaftaran siswa baru, tetapi fasilitasnya belum memadai, mungkin tahun berikutnya fasilitasnya sudah memadai.
3. Masalah yang ada yaitu Lambatnya dalam sistem penerimaan siswa baru seperti calon yang ingin mendaftar -

diperlukan waktu yang banyak, dalam sistem administrasi belum ada sistem untuk penyimpanan data, dikhawatirkan data yang telah di simpan hilang.

4. Syarat pendaftaran untuk penerimaan siswa baru :

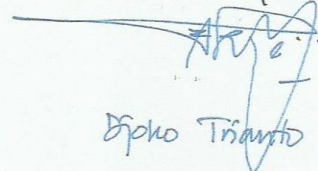
- a. Calon siswa memperlihatkan Ijazah Foto copy Legalisir 2 Lembar.
- b. Usia Maksimal 21 Tahun
- c. Calon siswa menerima dan mengisi form pendaftaran.
- d. Foto copy Ijazah/STTB yang telah di Legalisir 2 Lembar
- e. Foto copy SKL/~~SKKUN~~ SKHUN yang sudah di Legalisir 2 Lembar.
- f. Pas foto terbaru dan berwarna, ukuran $3 \times 4 = 4$ Lembar, $4 \times 6 = 2$ Lembar.
- g. Foto copy raport kelas 3 / Semester terakhir.

5. Proses seleksi ada 2 yaitu

- a. Untuk jenjang TK dan SD menggunakan batas umur sesuai dengan peraturan yang berlaku dan jumlah kuota yang tersedia.
- b. Untuk jenjang SMP dan SMK yaitu menggunakan batas nilai NEM atau SKHUN sesuai dengan kuota yang tersedia.

Jakarta, 01 April 2015

Tanda Tangan



Dpoko Trianto

Lampiran 04

Tabel Pengujian Kebutuhan Fungsional user

| No | Fungsi | Skenario Proses | Sistem Bekerja | Keterangan |
|----|-------------|-------------------------------------------------------------------------|----------------|------------------------|
| 1 | Menu | User membuka halaman beranda, daftar, masuk/login | ya | Benar |
| 2 | Create akun | User mengisi username, email, dan password dengan benar | ya | Benar |
| 3 | Create akun | User mengisi username saja pada form create akun | ya | Muncul Pesan Kesalahan |
| 4 | Create akun | User mengisi username dan email saja | ya | Muncul Pesan Kesalahan |
| 5 | Login | User login menggunakan username dan password yang benar | ya | Benar |
| 6 | Login | User melakukan login dengan username yang benar dan password yang salah | ya | Muncul Pesan Kesalahan |
| 7 | Login | User melakukan login dengan username yang salah dan password yang benar | ya | Muncul Pesan Kesalahan |

| | | | | |
|----|------------------------------|------------------------------------------------------------------------------------|----|------------------------|
| 8 | Login | User melakukan login dengan username yang salah dan password yang salah | ya | muncul pesan kesalahan |
| 9 | Lupa password | User lupa password dengan mengklik lupa password | ya | muncul notifikasi |
| 10 | Form pendaftaran | User mengisi dengan benar semua form pendaftaran yang disediakan lalu tekan simpan | ya | Benar |
| 11 | Form pendaftaran | User tidak mengisi bagian penting yang ada di form pendaftaran lalu tekan simpan | ya | muncul pesan kesalahan |
| 12 | Perbaharui data | user memperbaharui data diri yang sudah tersimpan | ya | Benar |
| 13 | Mencetak formulir pembayaran | User mencetak formulir pembayaran | ya | Benar |
| 14 | Mencetak biodata | User mencetak biodata yang sudah tersimpan | ya | Benar |
| 15 | Logout | User menekan menu logout | ya | Benar |

Tabel Pengujian Kebutuhan Fungsional admin

| No | Fungsi | Skenario Proses | Sistem Bekerja | Keterangan |
|----|-------------------------|--------------------------------------------------------------------------|----------------|------------------------|
| 1 | Login | admin login menggunakan username dan password yang benar | ya | Benar |
| 2 | Login | admin melakukan login dengan username yang benar dan password yang salah | ya | muncul pesan kesalahan |
| 3 | Login | admin melakukan login dengan username yang salah dan password yang benar | ya | muncul pesan kesalahan |
| 4 | Login | Admin melakukan login dengan username yang salah dan password yang salah | ya | muncul pesan kesalahan |
| 5 | Kelola Admin | Admin menambahkan akun baru dan menghapus akun | ya | Benar |
| 6 | Pendaftan | Admin melihat pendaftar yang sudah masuk | ya | Benar |
| 7 | Kelola data calon siswa | Admin mengupdate data calon siswa jika diperlukan dengan mengklik update | ya | Benar |

| | | | | |
|----|-------------------------------|-------------------------------------------------------------------------------------|----|-------|
| 8 | Kelola data calon siswa | Admin mendelete data calon siswa dengan mengklik delete | ya | Benar |
| 9 | Kelola data calon siswa | Admin melakukan registrasi pendaftar menjadi calon siswa dengan mengklik registrasi | ya | Benar |
| 10 | Kelola data calon siswa | Admin dapat merubah kembali status calon siswa menjadi pendaftar | ya | Benar |
| 11 | Kelola data calon siswa | Admin dapat melihat daftar calon siswa yang sudah falid di menu calon siswa | ya | Benar |
| 12 | Kelola data calon siswa | Admin dapat mengexport data calon siswa yang sudah falid kefile exel | ya | Benar |
| 13 | Kelola kelas | Admin dapat melakukan pembagian kelas | ya | Benar |
| 14 | Kelola data calon siswa | Admin dapat menghapus data calon siswa di menu calon siswa | ya | Benar |
| 15 | Kelola data calon siswa | Admin dapat mengupdate data calon siswa di menu calon siswa | ya | Benar |
| 16 | Kelola data calon siswa | admin dapat membuat pengumuman tentang penerimaan siswa baru | ya | Benar |
| 17 | Kelola pengumuman atau berita | Admin dapat mengupdate pengumuman | ya | Benar |

| | | | | |
|----|-------------------------------|--------------------------------------------------------------------------------------|----|-------|
| 18 | Kelola pengumuman atau berita | Admin dapat menghapus pengumuman | ya | Benar |
| 19 | Kelola fitur pendaftaran | Admin dapat mengatur fitur pendaftaran | ya | Benar |
| 20 | Kelola kuota | Admin dapat menentukan kuota calon siswa di setiap jenjang | ya | Benar |
| 21 | Kelola kuota | Admin dapat mengubah kuota di setiap jenjang | ya | Benar |
| 22 | Kelola seleksi | Admin dapat menentukan batas minimum nilai pada proses seleksi jenjang SMP dan SMK | ya | Benar |
| 23 | Kelola seleksi | Admin dapat merubah batas nilai minimum pada proses seleksi pada jenjang SMP dan SMK | ya | Benar |
| 24 | Kelola seleksi | Admin dapat menentukan batas umur minimum pada seleksi jenjang TK dan SD | ya | Benar |
| 25 | Kelola seleksi | Admin dapat merubah batas umur minimum pada seleksi jenjang TK dan SMP | ya | Benar |

| | | | | |
|----|--------------|------------------------------------------------------|----|-------|
| 26 | Lihat grafik | Admin dapat melihat grafik pendatar yang sudah masuk | ya | Benar |
| 27 | Logout | Admin mengklik Logout | ya | Benar |

Telah diuji

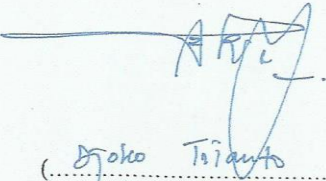
Hari/Tanggal :

Nama : Djoko Trianto

Jabatan : Ketua PSB

Tempat : YPI AS SALADAH

Tanda tangan


(Djoko Trianto)

TENTANG PENULIS

Mahfud Ramdhani Algifari, mahasiswa angkatan 2010 Pendidikan Teknik Informatika dan Komputer jurusan Teknik Elektro Universitas Negeri Jakarta. Lahir di Bekasi pada tanggal 24 Maret 1992. Pendidikan yang ditempuh oleh penulis adalah sebagai berikut : jenjang sekolah dasar lulus pada tahun 2004 di SDN 03 Kebalen. Selanjutnya jenjang Sekolah Menengah Pertama lulus pada tahun 2007 di SMPN 1 Bebelan. Jenjang Sekolah menengah atas lulus pada tahun 2010 di SMAN 1 Babelan.masuk perguruan tinggi pada tahun 2010 di Universitas Negeri Jakarta melalui jalur Mandiri (PEMABA). penulis telah menyelesaikan Skripsi yang berjudul “Pengembangan Aplikasi Penerimaan Siswa Baru di Yayasan Perguruan Islam Assa’adah Pondok Kelapa, Jaktim”.